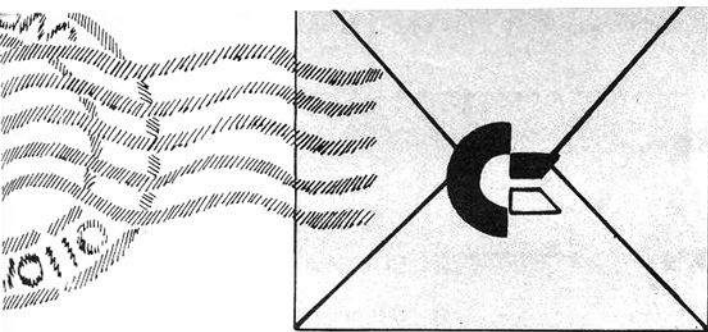
Commodore
COMPUTER CLUB
58
L. 4.500
La rivista degli utenti di sistemi Commodore
25 Novembre 1988 - Anno VII - N. 58 - Sped. Abb. Post. Gr. III/70 - CR - Distr.: Mepe - Tassa pagata per IP
Da questo mese
un mega spazio
per AMIGA
Il nome
del Kernal
ESCLUSIVO
Desk Top
Publishing:
anche col C64
C 16, viaggio
al centro
del Basic
MISSIONE
COMPIUTA
Come risolvere
le adventure dell'anno
systems
ALL'INTERNO LA CARTOLINA DEL
GRANDE CONCORSO
COMMODORE
DEE JAY

Sommario

Nella foto di copertina: Jennifer Jason Leigh dal film "Heart of Midnight" by Samuel Goldwin.

RUBRICHE

Direttore: Alessandro de Simone - **Caporedattore:** Michele Maggi
Redazione/collaboratori: Paolo Agostini, Davide Ardizzone, Claudio Baiocchi, Angelo Bianchi, Luigi Callegari, Sergio Camici, Umberto Colapicchioni, Maurizio Dell'Abate, Valerio Ferri, Roberto Ferro, Cristina Magnaghi, Giancarlo Mariani, Roberto Marigo, Clizio Merli, Marco Mietta, Marco Miotti, Oscar Moccia, Roberto Morassi, Guido Pagani, Antonio Pastorelli, Sonja Scharrer, Fabio Sorgato, Valentino Spataro, Danilo Toma
Grafica: Arturo Ciaglia, Elena Salvadori
Direzione, redazione, pubblicità: v.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Giorgio Ruffoni - Viale Famagosta, 75 - 20142 Milano Tel. 02/8467348
• Emilia Romagna: Spazio E - P.zza Roosevelt, 4 - 40123 Bologna - Tel. 051/236979
• Toscana, Marche, Umbria: Mercurio srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
• Lazio, Campania: Spazio Nuovo - via P. Foscari, 70 - 00139 Roma - Tel. 06/8109679
Segreteria: Tiziana Sodano - **Abbonamenti:** Liliana Spina
Tariffe: prezzo per copia L. 4.500. Abbonamento annuo (11 fascicoli) L. 45.000. Estero: il doppio.
Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 85.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizione: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: Systems Editoriale/La Litografica Srl - Busto Arsizio (Va)
Registrazioni: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70%
Distrib.: MePe - via G. Carcano, 32 - Milano
Periodici Systems: Banca Oggi - Commodore Club (disco) - Commodore Computer Club - Commodore Computer Club (disco produzione tedesca) - Computer - Computer disco - Electronic Mass Media Age -Energy Manager - Hospital Management - MondoRicambi - Nursing '90 - PC Programm (disco) - Personal Computer - Security - Software Club (cassetta ed. italiana) - VR Videoregistrare

la vostra posta

PROTESTE

□ Da molti mesi attendo inutilmente, dalla Commodore Italiana, l'invio del libretto di istruzioni in italiano per il mio Amiga 500.
(P. Petrini - Milano)
(Fabrizio Bassi - S. Gimignano)
(Sebastiano Terranova - Ravenna)

□ Da moltissimo tempo il mio drive 1581, nuovissimo ma non funzionante, giace presso un centro di assistenza ufficiale della Commodore Italiana e non c'è verso di riaverlo riparato.
(G. Caparotti - S. Giorgio CR)

• Non possiamo che rivolgere alla Commodore Italiana le lamentele segnalate, limitandoci a sperare che si tratti di semplici disguidi.

MPS/1000 PER AMIGA E 64

□ Posseggo un C/64 ed un modello di stampante della serie Mps Commodore (dotata, però, di libretto in tedesco). Potrò utilizzarla passando all'Amiga?
(Massimo Giacomello - Bologna)

• Se la stampante che possiedi è dotata di garanzia originale Commodore, dovrebbe esser sufficiente inviare alla Commodore Italiana una fotocopia della stessa per ottenere, gratuitamente, il libretto in italiano (salvo disguidi vari).

Per la compatibilità con l'Amiga tieni presente che i casi possono essere tre:

- Sul retro è presente il solo connettore seriale del tipo 1541: in questo caso la compatibilità è garantita solo per i piccoli Commodore (C/64, C/128, C/16 e così via).
- Se, invece, è presente il solo connettore per C/64 ma risulta praticabile la sostituzione di una cartuccia, è sufficiente procurarsi l'interfaccia idonea (a patto di trovarla) per collegarlo con l'Amiga.
- Se, infine, la stampante possiede due connettori (seriale Commodore e Centronics) la compatibilità è totale: a seconda dei casi potrai collegarla al C/64 oppure all'Amiga, ricordandosi di spegnere le apparecchiature prima di effettuare il collegamento.

MEGLIO COSI'

□ Vorrei segnalare l'efficacia di una piccola modifica da me apportata al programma "Grafici di funzioni per C/64" apparso sul n. 55 di C.C.C.
(Lelio Pacino - S. Ninfa)

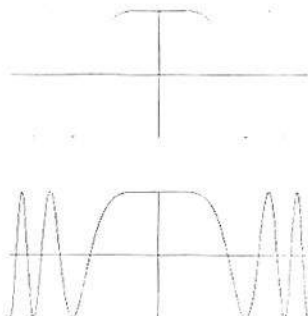
• "La modifica -prosegue il lettore- consiste nel sostituire la riga 400 con la seguente...

400 Xb = Mx* (I - X1): Yb = 20 + My* (Y2 - Fnf(i)): If I = X1 Then Xa = Xb: Ya = Yb

...e nell'aggiungere la seguente:

405 Line (Xa, Ya) - (Xb, Yb): Xa = Xb: Ya = Yb

In questo modo -conclude Lelio- vengono tracciati segmenti invece che singoli punti, con la coseguenza di migliorare la comprensibilità del grafico tracciato".

Ne approfittiamo per ricordare ai lettori che, molto più spesso di quanto non si creda, sono sufficienti piccole modifiche per migliorare programmi già sviluppati da altri.

SOLO SU DISCO

□ Seguo la vostra rivista da poco tempo e leggo che per digitare alcuni programmi è necessario prima caricare il Gw-Basic. Di che cosa si tratta e dove reperirlo?
(Anna Galgano - Portici)

• Il Gw-Basic Emulator, di cui parliamo nell'apposita rubrica, è un linguaggio, da noi realizzato, che consente al C/64 di seguire la sintassi del linguaggio interprete Gw-Basic, utilizzato sui personal computer IBM compatibili.

Ciò significa che, entro certi limiti, è possibile digitare (e veder girare) sul C/64, un programma scritto con l'identica sintassi di un PC.

Il Gw-Basic emulator (che si trova sullo stesso dischetto contenente il sistema operativo Ms-Dos emulator) è utilissimo soprattutto agli studenti che, possedendo "solo" un C/64, non potrebbero sviluppare a casa propria esercizi che, a scuola, hanno svolto sui computer Ms-Dos compatibili (IBM, Olivetti e così via).

Con una modica spesa, pertanto, chiunque possegga un C/64, dotato di drive, può sviluppare software che, digitato su un PC (o viceversa), genera gli stessi effetti sul più modesto C/64.

Il dischetto Ms-Dos emulator, contenente tutto il software necessario, può esser richiesto al nostro indirizzo, seguendo le indicazioni riportate in altra parte di questo stesso fascicolo.

Tieni presente che è possibile procurarselo solo su disco, proprio perchè il sistema operativo Ms-Dos (emulato, appunto, dal nostro software) non prevede minimamente l'uso del registratore.

FIDARSI?

□ Ultimamente non posso utilizzare il mio C/64 perchè alcuni tasti non funzionano più. Il mio negoziante afferma che è necessario sostituire il chip che gestisce la tastiera, ma non sono d'accordo con lui. Che cosa mi consigliate di fare?
(Lettera firmata - Montecatini)

• Se, premendo alcuni tasti, non viene generata la funzione desiderata, i motivi possono essere (almeno) tre.

E' possibile che sia entrata molta polvere nella stessa tastiera, inconveniente che impedisce il corretto contatto dei tasti; in questo caso, però, l'inconveniente dovrebbe scomparire battendo con decisione i vari tasti incriminati; per non esser costretti a pestare il computer (e provocar guai maggiori) come rimedio si potrebbe ricorrere ad un aspirapolvere, purchè non troppo potente, altrimenti i tasti verrebbero risucchiati(!).

L'inconveniente lamentato potrebbe essere attribuito anche ad un accidentale distacco parziale del connettore della tastiera dal circuito stampato, oppure ad un'ossidazione dei terminali del pettine. In entrambi i casi verrebbe impedito il contatto elettrico del connettore.

Oppure potrebbe aver ragione il tuo negoziante; in quest'ultimo caso, ovviamente, l'unica alternativa è quella di affidarsi ad un centro di riparazioni specializzato.

AIUTATECI A SERVIRVI MEGLIO

Spesso alcuni lettori, che dichiarano di possedere numeri arretrati del nostro periodico, porgono quesiti le cui risposte sono già state esplicitamente pubblicate (in occasione di risposte ad analoghe domande) oppure sono contenute in articoli presenti nei fascicoli in loro possesso.

Per evitare di ripetere argomenti già trattati, pertanto, ricordate di indicare sempre, nelle lettere che ci inviate, i numeri dei fascicoli in vostro possesso: potremmo infatti indicarvi, se esistono, gli articoli che, in un modo o nell'altro, possono chiarire gli argomenti richiesti.

Si ricorda ai lettori che non ci è possibile rispondere privatamente, nemmeno se si acclude l'affrancatura per la risposta.

COPIAR FILE O CAMBIAR COMPUTER

□ **Abituato al vecchio Commodore Pet, che consentiva di effettuare copie di Back-Up con un solo comando, trovo molte difficoltà a compiere la stessa operazione con il mio C/128 corredato sia di drive 1571 che di 1541. Volendo cambiar computer, inoltre, pensate che potrei utilizzare l'esperienza finora accumulata?**
(A. Angelo - Italia)

• Sul nostro dischetto "Directory" compaiono spesso utility di copia che consentono di effettuare le operazioni richieste. Del resto, contattando vari appassionati, non è difficile entrare in possesso di interi package di copiatura estremamente veloci ed affidabili.

Il nostro lettore, d'altra parte, ci informa (nella sua lunga lettera) che ha intenzione di acquistare due nuovi drive 1581 e penso, di conseguenza, che il problema sia da considerare sotto altro aspetto.

Quando si decide di operare nel campo "professionale", sarebbe opportuno considerare il lato economico, nel senso più generale. Mi riferisco, infatti, non solo al puro investimento di denaro ma, soprattutto, al tempo necessario per sfruttare al meglio le possibilità della macchina che si decide di acquistare.

Dalla decrizione del sistema posseduto dal nostro lettore, e dalla considerazione che i 1581 non si trovano (ancora) nei fustini del Dixan, arguisco che l'aspetto economico, pur se presente, non è quello principale.

Il mio consiglio spassionato è quindi quello di non continuare ad operare nel campo dei "piccoli" computer (quale è il C/128), ma di considerare con la massima attenzione calcolatori più potenti, che possono sicuramente dare maggiori soddisfazioni a chi intende salire di livello.

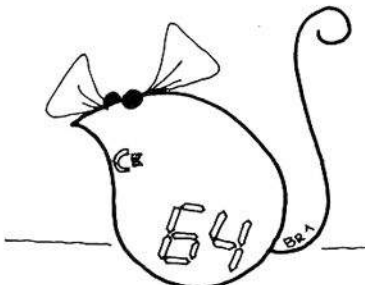
Quali computer, allora? Mi sembra ovvio: Amiga, oppure Ms-Dos compatibili (meglio se dotati di microprocessore 80286).

CARICAR DOLLARI

□ **Digitando: Load "$$ ",8 (cioè con due simboli del dollaro) viene caricata l'intestazione ed il numero di blocchi liberi presenti sul disco. Come mai?**
(Gianni Ronchi - Senago)

• La sintassi legata alla visualizzazione della directory consente di "estrarre" anche alcuni file dotati di particolari caratteristiche. Ad esempio con...
Load "Pip* ",8
...vengono caricati solo i nomi dei file, eventualmente presenti, che iniziano con PIP.

Con la sintassi del doppio dollaro, quindi, vengono caricati tutti i file che, come nome, hanno un dollaro, cioè... nessuno!

Ne approfittiamo per ricordare altre forme sintattiche poco conosciute. Con...
Load "$:*=p", 8
Load "$:*=s", 8
Load "$:*=r", 8
Load "$:*=u", 8
...vengono caricati, rispettivamente, solo i nomi dei programmi, dei file sequenziali, relativi, o user eventualmente presenti sul dischetto. Anche in questo caso, ovviamente, verranno visualizzate solo due righe in caso di assenza dei file cercati.

SCUOLA E LAVORO

□ **Avendo intenzione di abbandonare gli studi, sarei orientato a seguire uno di quei corsi di informatica in cui assicurano (anzi, promettono) l'inserimento immediato nel mondo del lavoro. Che cosa ne pensate?**
(S. M. - Sestri Levante)

• L'età minima della scuola dell'obbligo verrà presto portata a 16 anni; in quest'ottica è quindi assurdo pensare di abbandonare, oggi, gli studi della scuola superiore, per quanto noiosi possano sembrare e per quanto irraggiungibile possa apparire il traguardo.

Spesso il desiderio di avere qualche lira in tasca tutti i mesi porta molti giovani ad abbandonare gli studi per inseguire la chimera di un lavoro facile e ben pagato; le conseguenze di tale comportamento, purtroppo, si notano solo quando è troppo tardi per riprendere a studiare.

Se, quindi, non vi sono problemi economici in famiglia, che impediscono "materialmente" la prosecuzione degli studi, il mio consiglio è quello di sopportare pazientemente, e di conseguire questo benedetto diploma di maturità.

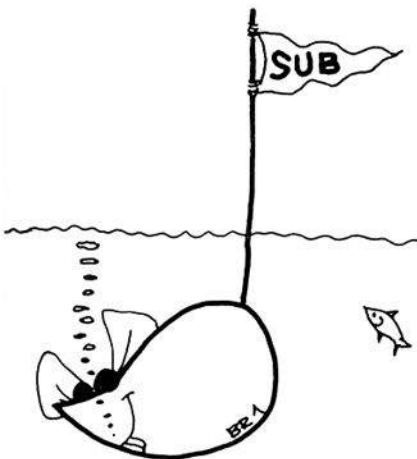
Se, invece, la decisione è irrevocabile, il mio consiglio è quello di avvicinare qualche studente che frequenti, o che abbia già frequentato, i corsi citati, in modo da formarsi un'opinione personale sull'affidabilità delle promesse e garanzie vantate nella pubblicità che compare sui manifesti che si vedono spesso in giro.

SODDISFATTO

□ **Sono davvero contento della Eprom per Mps-803, acquistata dalla Nuova Newel, e vorrei consigliarla ai lettori di C.C.C.**
(Gian Maria Calzolari - Parma)

• "Il nuovo set è stupendo -prosegue il nostro lettore- e stupisce il fatto che la stessa Commodore non abbia pensato ad inserire questa Eprom nello zoccolo predisposto. Con questo chip, inoltre, viene attivato l'ottavo ago della stampante che, altrimenti, rimarrebbe inutilizzato. C'è solo da segnalare che usando insieme Geos, Speed dos e Kit 803 possono verificarsi vari malfunzionamenti. In questi casi è però sufficiente accendere la stampante solo quando serve e spegnerla dopo averla utilizzata".

DA PLUS-4 A C/64

□ **Ho problemi nell'uso del programma da voi pubblicato su Personal**

Computer N. 8 che consente di "trasformare" un file prodotto dal word processor del Plus/4 in uno idoneo ad esser trattato dal w/p "Easy Script" per C/64.
Inoltre il mio C/128...
(Romolo Pranzetti - Vicarello)

• Il brevissimo programma segnalato non contiene errori di sorta.

Il motivo per cui, a conversione effettuata, ogni parola occupa un rigo intero, è dovuto, probabilmente, alla dimenticanza di qualche carattere di punto e virgola (;) la cui digitazione è indispensabile, ove indicato, per il "legame" tra parole successive.

Il nostro lettore segnala, inoltre, un malfunzionamento nel suo C/128. A volte lo schermo "scompare" o vengono evidenziati particolari disturbi che, però, scompaiono spostando la tastiera o pigiando la custodia del computer.

Il problema è sicuramente dovuto a qualche falso contatto che mi auguro circoscritto al cavetto del video. Prova ad inserirlo correttamente sia dal lato computer sia dal lato monitor oppure a sostituirlo. Se l'inconveniente dovesse persistere, il motivo è da attribuire quasi certamente a qualche saldatura fredda di uno (o entrambi) i connettori video (computer e/o monitor).

SOFTWARE

MINI TEXT EDITOR

□ Aiuto! Ho 12 anni e non riesco assolutamente a realizzare un banale Text Editor per usare la mia stampante come una macchina da scrivere.
3109(Riccardo Cascione - Rivarolo)

• Ecco il listato che chiedi:

```
100 rem demo mini text editor per c/64
110 :
120 print chr$(147)"accendi la stampante e digita la frase che vuoi: premendo
130 print "il tasto return la frase digitata verrà scaricata su stampante"
140 print "premendo shift + return si ritorna al programma principale"
150 print "questa routine ignora i tasti cursore e controllo schermo"
160 print "ma consente la cancellazione (tasto del)"
167 print: print chr$(18) "premi un tasto"
170 get a$: if a$="" then 170
180 gosub 17500
9998 :
9999 end
17500 x1=0: x2$="": close4: open4, 4, 7
17510 print chr$(14), chr$(8), chr$(147):
17520 print chr$(18)"  mini text editor per c/64 e mps/803  "
17530 print chr$(127);: poke211, peek(211)-1
17540 get x1$: if x1$="" then 17540
17550 x3=asc(x1$): if x3=141 then 17590: rem shift + return
17555 if x3=17 or x3=145 or x3=29 or x3=157 or x3=19 or x3=147 or x3=148 then 17540
17560 if x3=13 then x1=x1+1: print" ": print#4, x2$: x2$="": goto 17530
17564 if x3=20 and len(x2$) > 1 then x2$=left$ (x2$, len(x2$) -1): print x1$;: goto 17540
17565 if x3=20 and len(x2$) < 1 then 17540
17567 print x1$;: x2$=x2$ + x1$
17570 if x1=23 then x1=0: goto 17520
17580 goto 17530
17590 print #124: close4: print chr$(9), chr$(142), chr$(147): return
17599 rem mini text-editor per c/64 e stampante mps-803 e mps-801 compatibili
```

Mi permetto solo di ricordare che il listato è tratto, pari pari, dal n.42 di C.C.C. rubrica "Enciclopedia di Routine Basic".

```
100 REMDA PLUS 4 IN EASY SCRIPT
110 REMBY ALESSANDRO DE SIMONE
120 :
130 PRINTCHR$(5)CHR$(14)CHR$(147)"...................12345678901234567 6"
140 PRINTCHR$(14):INPUT"NOME FILE PLUS-4";A$:A$=A$+",S,R"
150 INPUT"NOME EASY SCRIPT";B$:B$=B$+",S,W"
160 OPEN1,8,2,A$:Y=1:PRINTCHR$(147);
170 OPEN8,8,8,B$:Y=1
180 GET#1,A$:IFA$=""THEN180
190 X=ASC(A$):IFSTTHENCLOSE1:CLOSE8:END
200 IFX=159THENPRINT:PRINT#8:Y=1:GOTO180
210 IFX=32ANDY=1THEN180
220 Y=0
230 IF(X>0)AND(X<27)THENPRINTCHR$(X+64);:PRINT#8,CHR$(X+64);:GOTO180
240 IFX>64ANDX<91THENPRINTCHR$(X+128);:PRINT#8,CHR$(X+128);:GOTO180
250 IF(X>26)AND(X<65)THENPRINTA$;:PRINT#8,A$;:GOTO180
260 GOTO180
```

SOFTWARE

"APPENDER" FILE

□ Sul n. 19 della vostra rivista è indicato un metodo per aggiungere dati ad un file sequenziale che, però non ho ben compreso.
(Claudio Masetti - Scandicci)

• Premettiamo che non è possibile realizzare la funzione Append con il registratore, ma solo con un floppy. Quando viene creato un file sequenziale su disco, la B.A.M. (che possiamo definire come una specie di "indice" usato dal computer) contiene una serie di puntatori per individuare, sul disco, le zone di memoria in cui sono memorizzati i dati che costituiscono il file stesso.
Nulla di più facile (per il computer) che individuare la fine dei dati ed aggiungerne altri.
Per illustrare meglio la procedura da seguire ricorriamo ad un esempio. Supponiamo di aver creato il file sequenziale dal nome "Prova 1" in cui sono stati memorizzati dieci numeri (da 1 a 10):

```
100 open 1,8,8,"prova 1,s,w"
110 for i=1 to 10
120 print#1, i: next
130 close 1
```

...in cui "w" sta per Write (= scrittura). Si noti che DEVONO realizzarsi le tre condizioni per la corretta creazione del file: apertura, scrittura, chiusura.
Supponiamo che, in seguito, si voglia aggiungere, in coda al file già realizzato (e dunque già presente su disco) un altro gruppo di numeri, ad esempio da 11 a 20. Il programma idoneo allo scopo non è più così "libero" come il primo; il nome, anzitutto, deve essere rigorosamente lo stesso (nel nostro caso, "prova 1") ed i suffissi devono essere indicati esplicitamente: "s" (per sequenziale) ed "a" (per Append). Esempio:

```
100 open 1,8,8,"prova 1,s,a"
110 for i=11 to 20
120 print#1, i: next
130 close 1
```

Per verificare che, in effetti, i nuovi valori siano stati aggiunti in coda, sarà sufficiente un programmino banale come il seguente:

```
100 open 1,8,8,"prova 1,s,r"
110 for i=1 to 20
120 input#1, i:print i:  next
130 close 1
```

...in cui "r" sta per Read (= lettura). E' bene sottolineare che i tre listati sono ridotti all'osso: è indispensabile aggiungere i vari controlli per verificare l'effettiva presenza del file su disco e, in lettura, sulla variabile ST per individuare la fine del file.

PLURILETTERA

□ Ho diverse domande da farvi (segue elenco).
(Nevio Limonata - Concorezzo)

• ed ecco le risposte:
1- Per entrare in possesso del libretto di istruzioni della tua stampante giapponese, l'unico sistema è quello di contattare l'importatore e chiedere delucidazioni in merito. Non ti illudere, comunque, di poter sfruttare tutte le sue caratteristiche con i programmi commercializzati per il C/64: non tutti questi, infatti, consentono di impostare il carattere speciale ESC, cioè Chr$(27), necessario per attivare particolari modalità di scrittura.

2- Abbiamo già detto diverse volte che non è possibile collegare due C/64 tra loro per mezzo del cavetto seriale; l'unica strada percorribile è la User Port.

3- Sul N. 53 di C.C.C. è comparso un intero articolo ("Se la memoria non vacilla") sulla selezione della Rom e della Ram del C/64.

4- Sul N. 56 un apposito articolo ("Lo strizzacervelli") illustra un metodo totalmente automatico per aggiungere nuovi comandi Basic.
Concludo suggerendo di abbandonare il registratore a cassette per passare ad un più efficiente drive per minifloppy.

128, CP/M E TURBO PASCAL

□ Sono soddisfatto perchè avete affrontato, finalmente, il Turbo-Pascal per il sistema operativo CP/M del mio C/128. Esiste una versione in grado di gestire anche la grafica del mio computer?
(Francesco Albano - Taranto)

• Nella (lunga) lettera il nostro lettore lamenta anche la mancanza di serietà dimostrata da un negoziante che non solo gli ha venduto la versione 1.0, ma gli ha proposto l'acquisto di una fantomatica versione "aggiornata" del Turbo Pascal non garantendo, però, la possibilità di attivare la grafica.
Come già ho avuto modo di scrivere, la versione Turbo-Pascal Borland (la 3.02), recensita sul N. 56, è relativamente limitata rispetto a quella che gira sui PC della serie Ms-Dos. In ogni caso si tenga presente che il linguaggio mi è stato fornito dalla stessa Borland che, per ovvi motivi, mi ha certamente fornito la versione più recente disponibile per il C/128. Non posso quindi pronunciarmi su altre versioni T. Pascal CP/M, dotate di ulteriori caratteristiche, sconosciute alla stessa Borland.
Il Turbo-Pascal per C/128, come già ho avuto modo di dire, rappresenta, in ogni caso, una valida alternativa per gli studenti che, dovendo studiare applicazioni di tale compilatore, non dispongono di un PC, nè hanno, almeno per ora, intenzione di acquistarne uno.

LA TERZA LOCAZIONE

□ Come posso spostare uno sprite oltre il 256-mo pixel orizzontale di schermo?
(Angelo Candelori - Avezzano)

• In una qualsiasi locazione è possibile memorizzare uno dei primi 256 numeri interi (numerati da 0 a 255). Poichè lo schermo è largo 320 pixel, il C/64 è costretto ad utilizzare, per la coordinata orizzontale due

locazioni; per quella verticale, infatti, è sufficiente un solo byte.

Supponiamo di operare con lo sprite N.0. In questo caso la prima delle due locazioni (ed esattamente la n. 53248) gestisce lo sprite dal pixel 0 al 255; la seconda, invece, (la n. 53264) interviene quando lo sprite deve superare la 256-ma posizione. Poichè è sufficiente un dispositivo di tipo semplice (sì no, 0 / 1, acceso / spento, e così via) per indicare il superamento, o meno, del 256-mo pixel, i progettisti della Commodore hanno pensato bene di non sprecare un'intera locazione di memoria per svolgere questa funzione ma, al contrario, di utilizzare una locazione per informare il calcolatore sulla situazione degli otto sprite contemporaneamente.

Pertanto ognuno degli otto bit che compongono il byte n. 53264 rappresenta (a seconda se posti a uno oppure a zero) lo "sconfinamento", del rispettivo sprite, al di là della posizione fatidica. In particolare, allo sprite n. 0 è associato il bit 0, al n. 1 il bit 1 e così via fino al settimo (che è... l'ottavo sprite).

Per spostare uno sprite oltre il 256-mo pixel è quindi necessario alterare lo stato del corrispondente bit della locazione 53264, tenendo presente che l'altra locazione preposta alla coordinata X (la 53248 nel caso dello sprite n. 0) rappresenta sempre una "parte" della stessa coordinata.

Se, ad esempio, il bit 0 di 53264 è posto a 1 e la locazione 53248 contiene il valore 80, lo sprite n. 0 verrà posizionato sul pixel n. 256 + 80, cioè 336. In linea teorica, pertanto, si potrebbe raggiungere il pixel n. 512 che, tuttavia, essendo al di là del bordo destro, risulterà invisibile all'operatore.

ANCORA SULLA LOCAZIONE 1

□ Studiando il disassemblato di alcuni giochi ed utility sofisticate per C/64, ho notato spesso l'alterazione della locazione n. 1. Quale è il motivo di questa tecnica?
(Enrico Girardi - Mogliano)

• Sul N. 53 di C.C.C. è comparso un intero articolo ("Se la memoria non vacilla") che illustra varie tecniche di selezione della Rom e della Ram del C/64. In pratica si tratta di tecniche particolari che consentono di sfruttare la Ram posta al di sotto del Basic e/o del Sistema Operativo del computer che, come è noto, sono memorizzati in Rom.

Per vari motivi, tra cui figurano le tecniche di protezione, i progettisti dei giochi allocano, sotto le Rom, intere parti di programma, schermate in alta risoluzione, dati relativi a brani musicali ed altro; al momento opportuno, alterando opportunamente la locazione 1, "estraggono" ed attivano il contenuto di tale area di memoria. Poichè la sola memoria gestibile al momento dell'accensione, nonostante tutto, risulta esigua per certe applicazioni, quasi tutti i programmi di una certa lunghezza fanno ricorso alla tecnica descritta, se non altro per non costringere l'operatore a cambiar spesso disco per sostituire musiche, schermate grafiche e subroutine di varia utilità.

MIGLIORIE

□ Vorrei, ma non ho il tempo di incrementare le mie conoscenze sul Basic (figuriamoci sul l.m.). Vi prego, pertanto, di indicare sempre le migliorie che suggerite di apportare ai vari programmi che pubblicate.
(Vincenzo Giampietro - Brindisi)

• Quando, nella "scheda tecnica" che spesso compare in calce agli articoli, suggeriamo ai lettori di apportare piccole modifiche (per adattare il software pubblicato ai vari calcolatori o per ottenere opzioni più potenti), ci riferiamo ad operazioni inevitabilmente generiche, che chiunque dovrebbe essere in grado di svolgere. Conosco già l'obiezione del lettore: "e se qualcuno non riesce, nonostante tutto?".

Purtroppo i suggerimenti, che spesso compaiono sulla rivista, si prestano ad interpretazioni erronee; sembrerebbe, infatti, che solo ove indicato siano possibili adattamenti, altrove no.

Nel campo dell'informatica, invece, è vero il contrario: su qualunque computer è possibile applicare le tecniche apprese e sfruttate su altri calcolatori, almeno per ciò che riguarda i principi fondamentali di funzionamento. Certo è impossibile, ad esempio, far apparire, sul C/64, tutti i colori di Amiga; però con un po' di pazienza (e tanta fantasia) qualcosa si riesce a fare. Analogamente è impraticabile la contabilità di un'azienda servendosi di un C/16, ma per uno studente di ragioneria (che dispone del solo C/16) non vi sono altre vie per esercitarsi nella programmazione.

Non dimentichiamo, infine, che un qualsiasi programma può esser portato alle estreme conseguenze. I moderni programmi di Desk Top Publishing (che consentono di realizzare, a casa propria, libri e giornali) hanno, come punto di partenza, i banali word processor che, esasperati fino all'inverosimile, offrono risultati che era impossibile immaginare qualche anno fa.

Allo stesso modo il programma Ge.Da.Fi. cui il lettore si riferisce nella lettera, può essere arricchito in modo pressochè infinito: lo ha dimostrato un nostro lettore (Giuseppe Mattencini di Bari, vedi C.C.C. n. 55) che lo ha adattato al C/128 in modalità 80 colonne.

La nostra pubblicazione, del resto, si rivolge anche ai principianti, ma con lo scopo di spingerli a fare sempre di più e meglio. Ritengo che seguire il "percorso" del programma specifico (Ge.Da.Fi.) sia una operazione certamente semplice, che non richiede lo studio del manuale del computer, ma solo (poco) tempo e tanta passione.

Non volendo (o non potendo) dedicare neanche un paio d'ore ad apportare migliorie (che, se conseguite, offrono soddisfazioni incredibili agli appassionati) non rimane che affidarsi ai programmi commercializzati che, pur se efficienti come (e più) di quelli offerti dalle riviste del settore, devono esser considerati come veri e propri strumenti di lavoro.

MANNAGGIA A LUI

□ Da quando un mio amico ha digitato, non so che cosa, sul mio Plus/4,

questo non riesce più a "colloquiare" con il drive.
(Alba R. Chianni)

• Non prendertela con il tuo amico, si tratta solo di una coincidenza. E' infatti impossibile, via software, danneggiare un computer, ed il mancato caricamento dei file dal drive può esser dovuto ad un accidentale distacco del cavetto seriale o ad altre cause difficili da diagnosticare a distanza. Una cosa è certa: il tuo amico non ha alcuna colpa, a meno che non abbia digitato qualcosa usando un martello invece delle dita...

CHE VUOL DIRE?

□ Sulla confezione del mio C/64 c'è scritto "Mit Geos". All'interno, invece, c'è solo il computer. E' normale tutto ciò?
(Cinzia Parenti - Roma)

• Le due parole tedesche "Mit Geos" significano "Con Geos". Alcune confezioni del C/64 (tra cui la tua) dovrebbero contenere un dischetto che consente, ai soli possessori di drive, di sostituire il sistema operativo originale con quello nuovo denominato, appunto, Geos. Questo consente di fare, con maggior semplicità e rapidità, varie operazioni con il C/64.

E' molto probabile che l'esemplare in tuo possesso provenga da canali "paralleli" di importazione; in ogni caso è chiaro che la confezione è stata manomessa prima di esser posta in vendita. Se questo è il tuo caso, purtroppo, non puoi usufruire del diritto di richiedere, alla Commodore Italiana, il libretto di istruzioni in italiano che desideri.

VIVAMIGA

□ Vorrei dare alcuni suggerimenti per migliorare i contenuti della rivista in modo da offrire, ai possessori di Amiga, una sezione tutta per loro.
(Simone Bernardi - Siena)

• Questa rappresenta certamente una lettera "storica" per Commodore Computer Club perchè, grazie ad essa, è stato superato il numero minimo di richieste che ci "costringono" ad affrontare, sistematicamente, nuovi argomenti.

Non si può comunque negare che, da parecchio tempo, Amiga trova spazio sulle nostre pagine. Da questo momento, tuttavia, verrà dedicato uno spazio specifico, con tanto di rubriche, posta ed altra "roba" di notevole interesse.

E' ovvio che avremo bisogno di un po' di tempo per raggiungere le condizioni di regime, ma è risaputo che le promesse siamo abituati a mantenerle.

Tutti gli amighisti, pertanto, stiano all'erta; tra poco tocca a loro!

(P.S.: invia pure quell'articolo di cui parli, giuro che lo leggerò con la massima attenzione).

DILEMMA

□ Da circa un anno un dilemma arrovella la mia mente: comprare un Amiga, un Ms-Dos o tutti e due assieme?
(Nunzio Santini - Comiso).

• Ahimè, duro è il responso! Io l'ho risolto con la terza soluzione, ma so benissimo che non tutti possono farlo.

Volendo procurarsi un modello Amiga si tenga conto che occorre anche il monitor a colori, l'espansione ad almeno 1 mega ed un drive aggiuntivo.

Nel caso si sia orientati verso un Ms-Dos, invece, sarebbe opportuno privilegiare i modelli basati sul microprocessore 80286 evitando i modelli più lenti (basati su 8088 oppure 8086) o non ancora sfruttati e carissimi (basati sull'80386).

Condire il tutto con la disponibiltà di denaro liquido e servire caldo.

COMPUTER ARCHITETTO

□ Dovendo risolvere strutture architettoniche per mezzo di matrici pluridimensionali, sono costretto ad abbandonare il C/64. Mi piace la grafica di Amiga, ma...
(Francesco Spagnuolo - Grottaglie)

• Alt! Se il problema da risolvere è di tipo professionale (destinato a produrre reddito, o vil denaro in genere), rimane valido un solo, unico, indifferibile ed immarcescibile consiglio: acquistare un calcolatore per il quale siano già disponibili pacchetti applicativi di cui si conosce l'efficacia, l'affidabilità e l'assistenza tecnica di seri(issim)e software house.

Solo se si desidera realizzare da soli il sofware specifico (a patto di esserne capaci) un qualsiasi computer può esser idoneo. Magari usando altri linguaggi, diversi dal "solito" Basic.

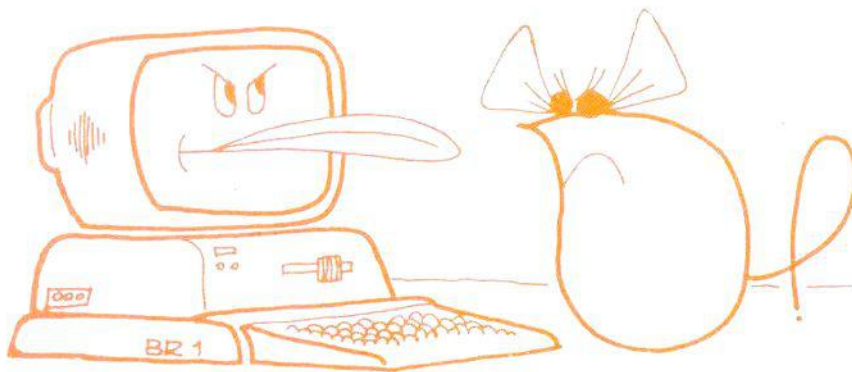
DIGITALIZZATORE VIDEO

□ Ho acquistato la scheda Provideo della Niwa (recensita sul N. 55 di C.C.C.) ma non riesco a far apparire alcuna immagine.
(Fedele Vignone - Napoli, ed altri lettori)

• Per far funzionare correttamente la scheda in questione è necessario che il segnale provenga dall'uscita video di un videoregistratore, di una telecamenra, di un sintonizzatore TV oppure di un moderno televisore, come è specificato, del resto, nell'articolo in questione.

Non tutti i televisori, infatti, offrono l'uscita video; collegare l'ingresso della scheda ad una comune antenna (centralizzata o esterna che sia) non sortisce alcun effetto.

Alcuni lettori, inoltre, sbagliano nell'effettuare i collegamenti: invece dell'uscita video, utilizzano l'ingresso. Molti apparecchi (soprattutto videoregistratori) sono ambi-

gui nella stampigliatura del pannello dei connettori. In corrispondenza del plug denominato "Video Out", infatti, spesso il costruttore intende indicare non il segnale in uscita, ma il luogo in cui inserire il connettore "Video Out" proveniente da una telecamera o apparecchio analogo. Altrettanto dicasi per "Video In".

COMMODORE A SCUOLA

☐ **Siamo orgogliosi di far presente che la nostra scuola (scuola media "Pescetti") partecipa al progetto per l'introduzione dell'informatica nella scuola dell'obbligo a Sesto Fiorentino; vengono usati, allo scopo, esclusivamente C/64, disk drive e stampanti Commodore. In tale sperimentazine sono coinvolti circa 1500 alunni, dei quali oltre 250 delle nostre classi.**
(Prof. Valerio Bandini - S.to Fiorentino)

• Per iniziare a parlare di informatica, infatti, un C/64 è più che sufficiente e, soprattutto, non appesantisce il già magro bilancio dell'istruzione pubblica italiana (chi ha orecchie per intendere...).

Vi consiglio di contattare direttamente la Commodore Italiana per ottenere informazioni su eventuali sconti praticati su ordinazioni di grosse quantità di materiale (forniture per scuole, enti, club e così via).

PIU' COLORI SUL C/128

☐ **Vorrei sapere se è possibile, con il C/128, gestire, per ciascun pixel, 4 colori da scegliere a piacere.**
(Ciro Nigri - Verona)

• Certo che è possibile, ma è necessario ricorrere alla tecnica multicolor, argomento sul quale ci siamo spesso intrattenuti e sul quale torneremo ancora.

C'è però un limite che non può essere evitato, nemmeno ricorrendo a sofisticate tecniche di programmazione in linguaggio macchina.

I quattro colori possono, infatti, essere scelti a piacere a patto che siano gli stessi in ciascuna delle 1000 aree di dimensioni 8 x 8 pixel in cui è suddiviso lo schermo hi-res. Pur se questa può sembrare una limitazione, si tenga presente che ben raramente si ha l'esigenza di utilizzare più colori in un'area così ristretta. In ogni caso, anche se si riuscisse nell'intento, la modesta definizione del video renderebbe indistinguibile la differenza di più colori diversi troppo vicini tra loro. Mettiamoci in testa che per superare certi livelli è necessario cambiar computer e monitor.

Consigli sul drive da acquistare non posso darli, per ovvi motivi che puoi intuire.

1571 E 1541

☐ **Ho intenzione di procurarmi il drive 1571. E' compatibile con il 1541?**
(Diego Capocchi - Capoliveri)

• Il 1571, per quello che mi risulta, è totalmente compatibile con il 1541. Quest'ultimo, inoltre, presenta caratteristiche di minor pregio rispetto al 1571. Il drive che intendi procurarti (purtroppo uscito di produzione) è molto più veloce del 1541 e consente, se programmato opportunamente, di formattare entrambe le facce del dischetto portando la sua capacità ad oltre 300 Kbyte.

Ti ricordo, infine, che il 1571 è il drive ideale per il C/128 che possiedi.

DA DISCO A NASTRO

☐ **Come posso registrare su nastro un programma che si trova su disco?**
(Andrea Salvatori)
(Luigi Ciccarelli - Sulmona)

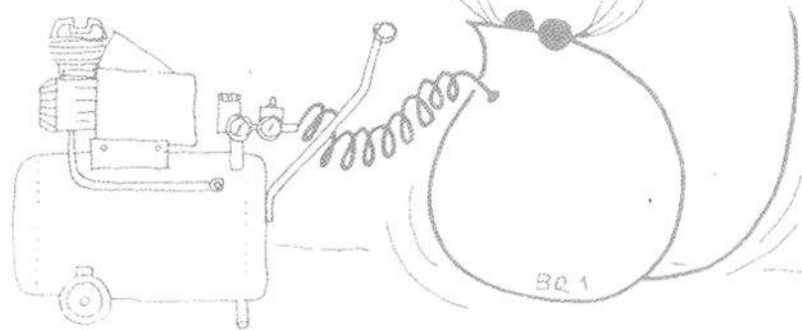
• Vi sono vari casi da considerare. Se il programma è un semplicissimo listato basic o, comunque, un programma che, da disco, si carica con un semplice...

Load "Nome", 8

...allora è sufficiente, dopo averlo caricato, registrarlo su nastro con un altrettanto banale...

Save "Nome"

Se, invece, è necessario caricarlo con la sintassi...

Load "Nome", 8, 1

...è indispensabile alterare i puntatori di inizio e fine memoria prima di dare il Save.

Se, poi, il programma "parte" appena viene caricato, la situazione è più difficile perchè bisogna individuare la tecnica di Autostart e rimuoverla o modificarla, sperando che il programma funzioni ancora dopo il trattamento.

Praticamente impossibile, invece, è l'adattamento di quei programmi che, durante il funzionamento, compiono numerosi accessi al disco oppure si caricano in più fasi successive.

Per risolvere questi problemi in modo totalmente automatico, sono in commercio numerose cartucce che permettono di riversare, su nastro o disco, qualunque programma di qualunque tipo.

Perchè non provi a telefonare alle Ditte che vendono tali cartucce per corrispondenza, il cui nome compare sulle pagine pubblicitarie della nostra rivista?

AUTORE DI ICONE

☐ **Terrei a far sapere che l'autore di alcune icone pubblicate sul dischetto Amigazzetta è PAOLO MARONCELLI e non Paolo Manardelli.**
(Paolo Maroncelli - Ravenna)

• Chiedo scusa per l'involontario disguido.

TOTIP

(Romano Demichelis)
Non ci è possibile sviluppare un programma specifico per il Totip perchè è stato... già fatto! Il package di programmi per il Totocalcio, infatti, può essere utilizzato anche per il Totip, l'Enalotto e per tutti i giochi che richiedono pronostici del tipo 1, 2, X.

risposte rapide

NON CONOSCO
(Claudio Belotti - Firenze)
(Luca Mineo - Palermo)

Purtroppo non ho mai usato la stampante che citi nella lettera e non posso dirti se è compatibile Mps/803.

DIETA
(Susanna Rossi - Empoli)

Il programma sulla dieta (C.C.C. n. 54) non contiene errori di sorta ed il motivo del malfunzionamento è dovuto sicuramente ad un'errata digitazione (forse del tipo indicato in fondo all'articolo stesso).

CARATTERI DISCENDENTI
(Filippo Staffa - Roma)

Per ottenere i caratteri discendenti sulla Mps/803 l'unica azione sensata è quella di sostituire la Rom. Per procurartela, ti consiglio di rivolgerti ai nostri inserzionisti la cui pubblicità compare sulle nostre pagine.

RICHIESTE GENERICHE
(Carmelo Giunta - Assoro)
(Sebastiano Tognacci - Venezia)
(Armando Oliva)

La domanda è troppo generica e non posso essere di alcun aiuto.

ELENCO BANCHE DATI
(Mario Ciciotti - Avezzano)

Purtroppo i titolari delle banche dati raggiungibili via modem sembrano esser gelosi dei propri archivi perchè non solo non fanno pubblicità sulle riviste di settore, ma non inviano nemmeno comunicati stampa (che verrebbero pubblicati gratuitamente) per informare sulle modalità da seguire per utilizzare i servizi che offrono.

FOTO SCONVENIENTI
(Enrico M. - Bari ed altri lettori)

Non mi pare che le foto di belle ragazze, da noi pubblicate di tanto in tanto, siano poi tanto erotiche e conturbanti. Terrò comunque presente la lamentela segnalata durante la selezione delle prossime immagini che, come si può già notare negli ultimi numeri, sono certamente più castigate.

HARD-COPY C/128
(Achille Tralli - Dosso)
(Alessandro Orsi - Torre del Lago)

La routine di hard copy pubblicata sul n. 52 non contiene errori di sorta; anzi, è proprio quella che utilizziamo di solito per riprodurre le schermate hi-res che appaiono sulla nostra pubblicazione.

DA C/16 AD AMIGA
(Massimiliano Baruffaldi - Chiavari)

Se non hai avuto difficoltà a realizzare ed usare software con il piccolo C/16, non ne avrai nemmeno con l'Amiga che intendi acquistare; semmai riuscirai più facilmente a scrivere programmi grazie alla notevole potenza del suo Basic.

APRIRE UN CLUB
(tre ragazzi di Palermo)

Per aprire un Club culturale (senza fini di lucro) non c'è bisogno di particolari permessi. Ti consiglio, tuttavia, di chiedere informazioni presso l'assessorato alla cultura della tua città o ad un qualsiasi commissariato di Polizia.

POCO INTERESSANTI
(Antonio Di Lorenzi - Napoli)

Per come sono stati sommariamente descritti, dubito che i programmi realizzati possano essere di un certo interesse perchè ve ne sono molti in circolazione che svolgono analoghe funzioni.

VECCHIO PET
(Aldo Cantavario, Via Bari 8/33, c.a.p. 16127 Genova)*

Il nostro lettore, che possiede un sistema Commodore PET completo, vuole contattare coloro che lo usano ancora.

NON ESISTE
(Erich Zamellato - Genova)

Non esiste un libro che contenga tutte le Poke del C/128.

IMPOSSIBILE
(Giorgio Vecchiocattivi - Sorso)

Il programma "La voce 3" può essere usato solo con il C/64 (o con il C/128 in modo 64) e non con altri computer o in modo 128.

MAI SU CARTA
(Simone Balestra - Codigoro)

Non possiamo prendere in considerazione articoli e programmi che non pervengano su disco.

TRA BREVE
(Giuseppe Franza - Foggia)

Tra breve verrà pubblicato un articolo sull'argomento richiesto.

SENZA ERRORI
(Valerio Marmiroli - Parma)
(Vincenzo Messina - Marano)

I listati citati nelle lettere non contengono errori. L'ultima versione delle routine di Toma è stata pubblicata sul fascicolo "Commodore speciale, linguaggio macchina".

MODIFICHE

(Raffaele Borrelli - S. Sebastiano)

Il circuito presentato sul N. 55 è suscettibile di numerose modifiche e personalizzazioni tra cui, ovviamente, anche quelle da te suggerite.

GIA' FATTO

(Federico Cassoli - Rovigo)

Più di una volta ci siamo intrattenuti sulla gestione del suono e della grafica del C/16. Ti consiglio di rileggere con attenzione i fascicoli di C.C.C. che dichiari di possedere.

PROSPETTIVA

(Enrico Rossitti - Mantova)

Hai ragione tu: la linea indicata nell'articolo citato non è quella di terra ma di orizzonte.

WINDOW

(Andrea Ravanelli - Imola)

Il modo di "aprire" una finestra sullo schermo del C/128 in alta risoluzione è descritto in modo semplice ed approfondito sul manuale di istruzione allegato al computer.

SOLO CIO' CHE VEDI

(Mario Sammartino - Capua)

Le pubblicazioni della Systems editoriale sono solo quelle citate nelle varie pagine pubblicitarie che appaiono (quasi) su ogni numero di C.C.C.

DAMA

(Giovanni Angini - Messina)

Non posso promettere di sviluppare l'argomento legato al gioco computerizzato della dama o di altri giochi da scacchiera.

STUDIARE

(Claudio Guidi - Cesena)

Non esiste un libro che tratta gli argomenti richiesti. Ti consiglio di digitare, studiare e modificare tutti i programmi che vedi pubblicati: è l'unico modo per acquisire esperienza.

NO

(Vito Catania - Catania)

Non è possibile ottenere altri stili di scrittura se il word processor adoperato non prevede specifici comandi per attivarli.

COLPISCI IL BERSAGLIO ...E VINCI!

Aut. Min. Ric.

CERCA LA CARTOLINA VINCENTE NEI COMMODORE POINT

DAL 14 NOVEMBRE AL 6 DICEMBRE

Tuta da Jogging colore bianco con marchio Commodore.

AMIGA 500
Personal Computer 16/32 BIT CPU 512K RAM. Incorpora un floppy disk drive da 3" 1/2 da 880 Kb e un mouse.

NUOVO C 64
Nuovo Personal Computer CPU 64K RAM. Computer ad alta risoluzione grafica, 256 combinazioni di colori, sintetizzatori di suono. Collegabile ad un qualsiasi televisore.

REGOLAMENTO

Sintonizzatevi su Radio DeeJay tutti i giorni dal lunedì al venerdì alle 17,40 e alle 18,40. Telefonate e partecipate al grande Concorso organizzato dalla vostra Radio preferita e da Commodore. Dettate al Disc Jockey le coordinate che pensate giuste per affondare la nave spaziale. Se ci riuscite vincete subito una delle splendide tute da jogging Commodore.
Altrimenti, in ogni caso, spedite la cartolina debitamente compilata e partecipate all'estrazione finale di: N. 1 Commodore Amiga 500; N. 4 Commodore C64 e degli altri premi eventualmente non distribuiti.

- Il Concorso si svolgerà dal 14/11 al 6/12/88 alle 17,40 e alle 18,40
- Le cartoline dovranno pervenire entro il 31/12/88
- L'estrazione avverrà entro il Gennaio '89.

Lombardia

Milano

- AL RISPARMIO - Viale Monza, 204
- BCS - Via Montegani, 11
- BRAHA ALBERTO - Via Pier Capponi, 5
- E.D.S. - Corso Porta Ticinese, 4
- E.S.C. - Via Roggia Scagna, 7
- FAREF - Via A. Volta, 21
- FLOPPERIA - Viale Monte Nero, 31
- GBC - Via Cantoni, 7 - Via Petrella, 6
- GIGLIONI LAURA - Via D'Ovidio, 8
- GILGLIONI - Viale Luigi Sturzo, 45
- LOGITEK - Via Golgi, 60
- MARCUCCI - Via F.lli Bronzetti, 37
- MELCHIONI - Via P. Colletta, 37
- MESSAGGERIE MUSICALI - Galleria del Corso, 2
- NEWEL - Via Mac Mahon, 75
- RIVOLA - Via Vitruvio, 43

Provincia di Milano

- IL CURSORE - Via Cavour, 35 - Novate Milanese
- REC ELGRA - Corso Milano, 118 - Bovisio Masciago
- F.LLI GALIMBERTI - Via Nazionale dei Giovi, 28/36 - Barlassina
- GBC - Viale Matteotti, 66 - Cinisello Balsamo
- P. GIORGIO OSTELLARI - Via Milano, 300 - Desio
- CASA DELLA MUSICA - Via Indipendenza, 21 - Cologno Monzese
- PENATI - Via Verdi, 28/30 - Corbetta
- EPM - Viale Italia, 12 - Corsico
- CENTRO COMPUTER PANDOLFI - Via Corridoni, 18 - Legnano
- COMPUTEAM - Via Vecellio, 41 - Lissone
- FUTURA - Via Solferino, 31 - Lodi
- M.B.M - Corso Roma, 12 - Lodi
- L'AMICO DEL COMPUTER - Viale Lombardia, 17 - Melegnano
- BIT 84 - Via Italia, 4 - Monza
- I.C.O. - Via dei Tigli, 14 - Opera
- NIWA HARD & SOFT - Via Bruno Buozzi, 94 - Sesto S. Giovanni
- FERRARI LUIGI - Via Madre Cabrini - Sant'Angelo Lodigiano

Brescia

- COMPUTER CENTER - Via Cipro, 62
- INFORMATICA 2000 - Via Stazione, 16/B
- MASTER INFORMATICA - Via F.lli Ugoni, 10/B
- VIGASIO MARIO - Portici Zanardelli, 3

Provincia di Brescia

- MISTER BIT - Via Mazzini, 70 - Breno
- CAVALLI PIETRO - Via 10 Giornate, 14/B - Castrezzato
- VIETTI GIUSEPPE - Via Milano, 1/B - Chiari
- MEGABYTE - Piazza Malvezzi, 14 - Desenzano del Garda
- CO-RE - Via XXV Aprile, 136 - Flero
- BARESI RINO & C. - Via XX Settembre, 7- Ghedi
- INFO CAM - Via Provinciale, 38 - Gratacasolo

Como

- IL COMPUTER - Via Indipendenza, 90
- 2M ELETTRONICA - Via Sacco, 3

Provincia di Como

- ELTRONGROS - Via L. da Vinci, 54 - Barzanò
- EGA - Via Mazzini, 42 - Cassago Brianza
- DATA FOUND - Via A. Volta, 4 - Erba
- EGA - Via A. Moro, 17 - Galbiate
- RIGHI ELETTRONICA - Via G. Leopardi, 26 - Olgiate Comasco
- CIMA ELETTRONICA - Via Leonardo da Vinci, 7 - Lecco
- FUMAGALLI - Via Cairoli, 48 - Lecco

Cremona

- MONDO COMPUTER - Via Giuseppina, 11/B
- PRISMA - Via Buoso da Dovara, 8
- TELCO - Piazza Marconi, 2/A

Provincia di Cremona

- ELCOM - Via IV Novembre, 56/58 - Crema
- EUROELETTRONICA - Via XX Settembre, 92/A - Crema

Mantova

- TUBALDO ELSO & C. - Galleria Fermi, 7
- 32 BIT - Via Cesare Battisti, 14
- CRESPI G. & C. - Viale Lombardia, 59 - Castellanza
- COMPUTER SHOP - Via A. da Brescia, 2 - Gallarate
- LIMA IMPORT-EXPORT - (Grandi Magazzini Bossi) - Via Clerici, 196 - Gerenzano
- J.A.C. - Via Matteotti, 38 - Sesto Calende

Valle d'Aosta

Aosta

- F.lli GATTI - Via Festaz, 75

Piemonte

Alessandria

- BIT MICRO - Via Mazzini, 104
- SERVIZI INFORMATICI - Via Alessandro III, 47

Provincia di Alessandria

- S.G.E. ELETTRONICA - Via Bandello, 19 - Tortona

Asti

- RECORD - Corso Alfieri, 166/3

Cuneo

- ROSSI COMPUTER - C.so Nizza, 42
- STUDIO SOFTWARE - C.so Nizza, 49

Provincia di Cuneo

- PUNTO BIT - Corso Langhe, 26/C - Alba
- SDI - Via Vittorio Emanuele, 250 - Bra
- ASCHIERI GIANFRANCO - Corso Emanuele Filiberto, 6 - Fossano

Novara

- ELCOM - Corso Mazzini, 11
- PROGRAMMA 3 - Viale Buonarroti, 8
- PUNTO VIDEO - Corso Risorgimento, 39/B

Provincia di Novara

- MIRCO POLACCO & C. - Via Monte Zeda, 4 - Arona
- ALL COMPUTER - Corso Garibaldi, 106 - Borgomanero
- MICROLOGIC - Via Giovanni XIII, 2 - Domodossola
- ELLIOT COMPUTER - Via Don Minzoni, 32 - Intra

Torino

- ABA ELETTRONICA - Via C. Fossati, 5/P
- BIT INFORMATICA - Via V. Emanuele, 154 - Ciriè
- HI-FI CLUB - Corso Francia, 92/C - Collegno
- I.C.S. - Stradale Torino N. 73 - Ivrea
- BAS - Corso Roma, 47 - Moncalieri
- CERUTTI MAURO - Corso Torino, 234 - Pinerolo
- EUREX - Corso Indipendenza, 5 - Rivarolo C.se
- DIAM INFORMATICA - C.so Francia, 146/Bis - Rivoli
- FULLINFORMATICA - Via Vittorio Veneto, 25 - Rivoli
- GAMMA COMPUTER - Via Cavour 3 A/B - Settimo Torinese

Vercelli

- ELETTROGAMMA - C.so Bormida, 27
- ELETTRONICA di BELLAMO A. & C. - Strada Torino, 15

Provincia di Vercelli

- C.S.I. TEOREMA - Via Losana, 9 - Biella
- SIGEST - Via Bertodano, 8 - Biella
- REMONDINO FRANCO - Via Roma, 5 - Borgosesia
- FOTOSTUDIO TREVISAN - Via XXV Aprile, 24/B - Cossato
- STUDIO FOTOGRAFICO IMARISIO - Piazza Martiri Libertà, 7 - Trino

Liguria

Genova

- ABM COMPUTER - Piazza De Ferrari 24/R
- CENTRO ELETTRONICA - Via Chiaravagna 10/R - Località Sestri Ponente
- COMMERCIALE SOTTORIPA - Via Sottoripa, 115/117
- FOTOMONDIAL - Via del Campo, 3-5-9-11-13/R
- LA NASCENTE - Via San Luca, 4/1
- RAPPR - EL - Via Boreoratti, 23/R

Imperia

- CASTELLINO - Via Belgrano, 44
- SASA COMPUTER - Via Nazionale, 256

Provincia di Imperia

- CENTRO HI-FI VIDEO - Via della Repubblica, 38 - Sanremo

- IL COMPUTER SERVICE SHOP - Via Padana Superiore, 197 - Vimodrone

Bergamo

- COMIF - Via Autolinee, 10
- CORDANI - Via dei Caniana, 8
- D.R.B. - Via Borgo Palazzo, 65
- NEW SYSTEMS - Via Paglia, 36

Provincia di Bergamo

- BERTULEZZI G. - Via Fantoni, 48 - Alzano Lombardo
- COMPUTER TEAM - Via Verdi, 1/B - Carvico
- OTTICO ROVETTA - Piazza Garibaldi, 6 - Lovere
- A.I.S. INTERNATIONAL - Via San Carlo, 25 San Pellegrino Terme
- SISTHEMA - Via Roma, 45 - Sarnico
- COMPUTER POINT - Via Lantieri, 52 - Sarnico
- AB INFORMATICA - Statale Cremasca, 66 - Urgnano
- ELETTRONICA di BASSO - V.le Risorgimento, 69

Pavia

- POLIWARE - Corso Carlo Alberto, 76

Provincia di Pavia

- LOGICA INFORMATICA - Via Monte Grappa, 32 - Vigevano
- M. VISENTIN - C.so Vitt. Emanuele, 76 - Vigevano

Sondrio

- CIPOLLA MAURO - Via Tremogge, 25

Provincia di Sondrio

- FOTONOVA - Via Valeriana, 1 - San Pietro di Berbenno

Varese

- DIMECO SISTEMI - Via Garibaldi
- IL CENTRO ELETTRONICO - Via Morazzone, 2
- SUPERGAMES - Via Carrobbio, 13

Provincia di Varese

- BUSTO BIT - Via Gavinana, 17 - Busto Arsizio
- ALEX COMPUTER - Corso Francia, 333/4
- COMPUTER HOME - Via San Donato, 46/D
- COMPUTING NEWS - Via Marco Polo, 40/E
- DE BUG - Corso Vitt. Emanuele II, 22
- DESME UNIVERSAL - Via San Secondo, 95
- F.D.S. - Via Borgaro, 86/D
- INFORMATICA ITALIA - Corso Re Umberto, 128
- MT INFORMATICA - Corso Giulio Cesare, 58
- MUSIC'S SHOP - Corso Potenza, 177
- NEW BUSINESS COMPUTER - Via Nizza, 45/F
- PLAY GAMES - Via C. Alberto, 39/A
- RADIO TV MIRAFIORI - C.so Unione Sovietica, 381
- SMT ELETTRONICA - Via Bibiana, 83/B
- TELERITZ - Corso Traiano, 34

Provincia di Torino

- PAUL E CHICO VIDEOSOUND - Via Vittorio Emanuele, 52 - Chieri
- CASTELLINO - Via Genova, 48 - Ventimiglia

La Spezia

- CATTONI - Via Vitt. Veneto, 75
- I.L. ELETTRONICA - Via Vitt. Veneto, 123

Provincia di La Spezia

- I.L. ELETTRONICA - Via Aurelia, 299 - Fornola di Vezzano

Savona

- CASTELLINO - Corso Tardy e Benech, 101
- ATHENA - Via Carissimo E. Crotti, 16/R

Veneto

Belluno

- UP TO DATE - Via Vittorio Veneto, 43

Provincia di Belluno

- GUERRA COMPUTERS - Viale Mazzini, 10/C - Feltre

Padova

- BIT SHOP - Via Cairoli, 11

- COMPUMANIA
  - Riviera Tiso Camposanpiero, 37
- COMPUTER POINT - Via Roma, 63
- D.P.R. - V.le Lombardo, 4
- GIANFRANCO MARCATO
  - Via Madonna della Salute, 51/53
- ZELLA ADELIO
  - Piazza De Gasperi, 31/A

Provincia di Padova
- CAERT - Via Andorra, 11
  - Zona Industriale Camin

Treviso
- BIT 2000 - Via Brandolini d'Adda, 14

Provincia di Treviso
- DE MARIN - Via XX Settembre, 74
  - Conegliano
- SIDESTREET - Via Salvo d'Acquisto, 8
  - Montebelluna
- FALCON - Via Terraggio, 116
  - Preganziol

Venezia
- TELERADIO FUGA
  - San Marco, 3457

Provincia di Venezia
- GUERRA EGIDIO & C.
  - Via Bissuola, 20/A - Mestre
- TREKILOWATT
  - Via Torre Belfredo, 47 - Mestre
- REBEL - Via F. Crispi, 10
  - San Donà di Piave
- GUERRA COMPUTERS
  - Via Vizzotto, 29
  - San Donà di Piave
- TELFERT - Via Chiesa, 1509
  - Sottomarina
- RADIOCESTARO - Via Roma, 89
  - Spinea

Verona
- CASA DELLA RADIO - Via Cairoli, 10
- TELESAT - Via Vasco de Gama, 8

Provincia di Verona
- FERRARIN - Via dei Massari, 10
  - Legnago

Vicenza
- ELETTRONICA BISELLO
  - Viale Trieste, 427/429
- SCALCHI MARKET - Via Cà Balbi, 139

Provincia di Vicenza
- GUERRA COMPUTERS
  - Via Dell'Industria - Alte Ceccato
- SCIAVOTTO - Via Zanella, 21
  - Cavazzale
- R.T. SYSTEM - Via L. da Vinci, 99

Provincia di Udine
- IDRENO MATTIUSSI & C.
  - Via Liciniana, 50 - Tavagnacco

Rovigo
- CLINICA DEL RASOIO E DEL COMPUTER
  - Via Fiume, 31/33

Trentino Alto Adige

Bolzano
- C.M.B. ITALIA - Via Roma, 82
- MATTEUCCI PRESTIGE
  - Via Museo, 54

Provincia di Bolzano
- ELECTRO TAPPEINER
  - P.za Principale, 90 - Silandro
- RADIO MAIR - Via Centrale, 70
  - Brunico
- ELECTRO RADIO HENDRICH
  - Via delle Corse, 106 - Merano

Trento
- CRONST - Via Galilei, 25

Emilia Romagna

Piacenza
- COMPUTER LINE - Via G. Carducci, 4
- DELTA COMPUTER
  - Via Martiri della Resistenza, 15/G
- SOVER - Via IV Novembre, 60

Toscana

Arezzo
- DELTA SYSTEM - Via Piave, 13

Firenze
- ATEMA
  - Via Benedetto Marcello, 1a/1b
- COOPERATIVA L.D.T. - Via Icaro, 9
- ELETTRONICA CENTOSTELLE
  - Via Centostelle, 5/a
- HELP COMPUTER
  - Via degli Artisti, 15/A
- PUNTO SOFT - Via Vagnetti, 17
- TELEINFORMATICA TOSCANA
  - Via Bronzino, 36

Provincia di Firenze
- WAR GAMES
  - Via Raffaello Sanzio, 126/A
  - Empoli
- NEW E.V.M. COMPUTER
  - Via degli Innocenti, 2
  - Figline Valdarno

Pisa
- C.H.S. - Via Carlo Cattaneo, 90/92
- ELECTRONIC SERVICE
  - Via della Vecchia Tranvia, 10
- IT - LAB - Via Marche 8A/8B

Pistoia
- ELECTRONIC SHOP
  - Via della Madonna, 49
- OFFICE DATA SERVICE
  - Galleria Nazionale, 22

Provincia di Pistoia
- ZANNI & C. - Corso Roma, 45
  - Montecatini Terme

Siena
- VIDEO MOVIE - Via Garibaldi, 17

Provincia di Siena
- ELECTRIC SHOP - Via A. Casini, 51
  - Chianciano Terme
- ELETTRONICA
  - Via di Gracciano nel Corso, 111
  - Montepulciano

Umbria

Provincia di Perugia
- COMPUTER STUDIOS
  - Via IV Novembre, 18/A
  - Bastia Umbra

Campania

Provincia di Avellino
- FLIP FLOP - Via Appia, 68 - Atripalda

Benevento
- E.CO. INFORMATICA
  - Via Pepicelli, 21/25

Caserta
- O.P.C. - Via G.M. Bosco, 24

Provincia di Caserta
- M.P. COMPUTER - Via Napoli, 30
  - Maddaloni

Napoli
- BABY TOYS
  - Via Cisterna dell'Olio, 5/Bis
- CASA MUSICALE RUGGIERO
  - Piazza Garibaldi, 74
- CENTRO ELETTRONICO CAMPANO
  - Via Epomeo, 121
- CI.AN - Galleria Vanvitelli, 32
- DARVIN - Calata San Marco, 26
- FALCO ELETTRONICA
  - Via Sarno, 100 - Striano
- TECNOTRE - Via P. Fusco, 1/F
  - Torre Annunziata

Salerno
- COMPUTER MARKET
  - C.so Vitt. Emanuele, 23

Provincia di Salerno
- KING COMPUTER - Via Olevano, 56
  - Battipaglia
- DIMER POINT - Via C. Rosselli, 20
  - Eboli

Puglia

Bari
- ARTEL - Via G. d'Orso, 9
- COMPUTER'S ARTS
  - Viale Meucci, 12/B

Provincia di Bari
- F. FAGGELLA - Corso Garibaldi, 15
  - Barletta
- G. FAGGELLA
  - Via P. d'Aragona, 62/A - Barletta
- G. LONUZZO - Via Nizza, 21
  - Castellana

Brindisi
- MARANGI & MICCOLI
  - Via Prov. San Vito, 165

Provincia di Foggia
- IL DISCOBOLO - Via T. Solis, 15
  - San Severo

Lecce
- BIT
  - Via 95° Regg.to Fanteria, 87/89

Provincia di Lecce
- CEDOK INFORMATICA - Via Roma, 31
  - Tricase

Matera
- GAUDIANO ELECTRONICS
  - Via Roma, 1

Taranto
- ELETTROJOLLY - Via de Cesare, 13
- TEA - Via Regina Elena, 101

Calabria

Catanzaro
- C. & G. COMPUTER - Via F. Acri, 28
- PAONE SAVERIO - Via F. Acri, 93/99

Provincia di Catanzaro
- COMPUTER HOUSE - Via Bologna
  - Crotone

point Commodore®

- COMPUTER B. COSTO
  - Via del Costo, 34 - Thiene
- ELETTROCASA - Via Roma, 67
  - Trissino

Friuli Venezia Giulia

Gorizia
- E.C.O. ELETTRONICA
  - Via F.lli Cossar, 23

Pordenone
- RIGO - Viale Cossetti, 5

Provincia di Pordenone
- MDT - Piazza Repubblica, 5
  - Villanova di Prata
- BRUNO DA PIEVE
  - Via Colombera, 17 - Porcia

Trieste
- AVANZO GIACOMO
  - Piazza Cavana, 7
- COMPUTER SHOP - Via P. Reti, 6
- COMPUTIGI - Via XX Settembre, 51
- CTI - Via Pascoli, 4

Udine
- MOFERT 2 - Via Leopardi, 21
- COSCI F.LLI - Via Roma, 26 - Prato
- CENTRO INFORMATICA
  - Via F.lli Cervi, 21/29 - Pontassieve

Grosseto
- COMPUTER SERVICE
  - Piazza Ponchielli, 2

Livorno
- ETA BETA - Via San Francesco, 30
- FUTURA 2 - Via Cambini, 19

Provincia di Livorno
- ELETTRONICA ALESSI PAOLO
  - Via Cimarosa, 1 - Piombino

Provincia di Lucca
- IL COMPUTER - Viale Colombo, 216
  - Lido di Camaiore
- SANTI VITTORIO - Via Roma, 23
  San Romano Garfagnana

Massa
- EURO COMPUTER
  - Piazza Bertagnini, 4
- FIRMWARE - Via Aurelia Ovest, 27

Provincia di Massa
- RADIO LUCONI - Via Roma, 24/B
  - Carrara
- ELETTRONICA RO.DA.LO.
  - Via Epomeo, 216/B
- GIANCAR 2 - Piazza Garibaldi, 37
- GRUPPO BUSH
  - Galleria Umberto I, 55
- ODORINO - Largo Lala, 22/A-B
- R 2 - Via F. Cilea, 285
- SPY - Via Fontana, 135
- TOP - Via S. Anna dei Lombardi, 12
- VIDEOFOTOMARKET
  - Via S. Brigida, 19

Provincia di Napoli
- SPADARO - Via Romani, 93
  - S. Anastasia
- TUFANO - S.S. Sannitica, 87 Km 7
  - Casoria
- ELETTRONICA 2000
  - Corso Durante, 40
  - Frattamaggiore
- GATEWAY - Via Napoli, 68
  - Mugnano
- NUOVA INFORMATICA SHOP
  - Via Libertà, 185/191 - Portici
- BASIC COMPUTER
  - C.so Garibaldi, 34
  - Pozzuoli
- OTTICA FOTO NELLO RUELLO
  - C.so Vittorio Emanuele, 177
  - Vibo Valentia

Cosenza
- SIRANGELO COMPUTER
  - Via N. Parisio, 25

Provincia di Cosenza
- ELIGIO ANNICHIARICO & C.
  - Via Roma, 21 - Castrovillari
- ALFA COMPUTER
  - Via Nazionale, 341/A
  - Corigliano Scalo
- ING. FUSTO SALVATORE
  - Corso Nicotera, 99
  - Lamezia Terme

Reggio Calabria
- CONTROL SYSTEM
  - Via S. Francesco da Paola, 49/DE
- SYSTEM HOUSE
  - Via Fiume ang. Palestino, 1

Provincia di Reggio Calabria
- COMPUTER SHOP
  - Via Matteotti, 50/52 - Locri

Elenco aggiornato al 19-10-1988

A tutto disco.

*Finalmente, viste le numerose richieste, d'ora in poi ogni pubblicazione **Software Club** su cassetta sarà disponibile anche su dischetto da richiedersi, per corrispondenza, presso la redazione.*

Sono disponibili i seguenti titoli:

Software Club #11 *(C/64-128, C/16, Plus/4 e Vic 20)*
Software Club #12 *(C/64-128, C/16, Plus/4 e Vic 20)*
Software Club #13 *(C/64-128, C/16, Plus/4 e Vic 20)*
Software Club #14 *(C/64-128, C/16, Plus/4 e Vic 20)*

I Gialli Commodore *(C/64-128)*
I Gialli Commodore 2 *(C64/C128)*
Charlie Deus *(C/64-128)*
La voce III *(C/64-128)*
Commodore 64 Club *(C64)*

Il prezzo, per ognuna delle suddette pubblicazioni è di L. 12.000 più L. 3.000 per spese di spedizione.

Le richieste vanno indirizzate a:

Systems Editoriale
Viale Famagosta, 75
20142 MILANO
Tel. 02/8467348

Il pagamento può essere effettuato tramite assegno bancario o versamento sul c/c N. 37952207
Non è possibile inviare materiale in contrassegno nè contro invio di vaglia telegrafico.
Per ogni ordine, anche se per più dischetti, le spese rimangono fissate in L. 3.000

Sempre un passo avanti.

UN VIRUS A SCOPPIO RITARDATO

Vediamo come è possibile creare un virus che si scateni solo dopo un ben definito periodo di "incubazione"

di **Michele Maggi**

Tra le caratteristiche che fanno del virus un pericoloso nemico ce n'è una in particolare che lo rende ancor più insidioso.

Si tratta dello "scoppio ritardato" che permette, all'ignaro utilizzatore, di agire come se il programma funzionasse perfettamente senza accorgersi che, in realtà, il programma tiene conto del numero di utilizzazioni.

Naturalmente, oltre un certo limite, scatterà l'infezione e... saranno guai!

Nel numero scorso abbiamo visto come realizzare un programmino in interrupt, a scopo puramente dimostrativo, per impadronirci delle tecniche di programmazione più adatte. Questa volta faremo di più, vedremo infatti come è possibile creare un virus resistente al Run / Stop e Restore e anche all'eventuale pressione del tasto Reset che, in teoria, dovrebbero eliminare qualsiasi interrupt riportando il computer alle condizioni standard.

Non ci credete? Continuate a leggere allora!

SCHEDA TECNICA

Software di protezione insolita per applicazioni generali.

Hardware richiesto: C/64; adattabile ad altri computer Commodore

Richiede il disk drive

Consigliato agli esperti

Anche il programma pubblicato in queste pagine è contenuto nel disco "Directory" di questo mese.

DISABILITANDO S'IMPARA

Parliamo per ora del tasto Restore: come è noto vi sono varie poke per disabilitarlo o dirottarlo (808, 225 ad esempio oppure 808, 234, che non blocca TI$) ma nel nostro caso il fatto di lasciarlo APPARENTEMENTE in funzione può tornare a nostro vantaggio. Vediamo perchè.

Qualunque utente sessantaquattrista, in caso di panico, preme Run Stop e Restore.

Ciò avviene quando il computer si "inchioda" oppure quando, magari tramite Peek e Poke, si sono spostati puntatori, banchi di memoria e così via.

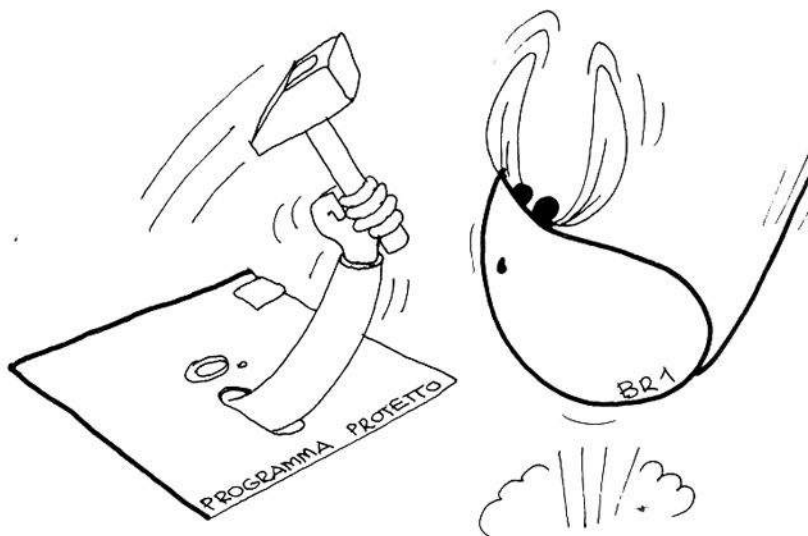
L'utente "comune", quindi, al primo sospetto di presenza di virus, sarà indotto istintivamente a premere i suddetti tasti, con la speranza di riportare tutto alla normalità; lo spegnimento del computer viene infatti considerato troppo drastico, se non dannoso, e non permetterebbe, in ogni caso, di copiare e/o modificare il programma presente in memoria.

Se il Run Stop Restore risulta inibito, l'utente, conscio del pericolo, sarà costretto a rimedi più efficaci, come il Reset o lo spegnimento, e i danni inflitti dal nostro virus potrebbero risultare limitati.

Perchè, allora, non far credere all'inconsapevole utente che il Run Stop Restore ha funzionato perfettamente, sconfiggendo il virus?

Questo trucco renderà l'utilizzatore pirata assai più vulnerabile: tranquillizzato, infatti, dalla consueta schermata vuota (ad eccezione del "ready"), si concederà qualche minuto di relax e sarà poprio questo a dare al nostro virus il tempo di agire portando a termine il suo compito.

Lo stesso discorso vale, a maggior ragione, per il Reset di sistema, considerato da tutti il rimedio più efficace.

Anche in questo caso, sia per il Reset hardware (con apposito pulsante) che per il Reset software (SYS 64738), l'operazione di re-inizializzazione verrà apparentemente eseguita, ma il virus sarà all'erta, più pericoloso che mai...

La realizzazione di tutto ciò presuppone una discreta conoscenza del Sistema Operativo in quanto sarà necessario fare riferimento ad alcune sue routine.

La prima cosa da sapere è che, in condizioni normali, non appena viene premuto Restore o si verifica un Reset (hard o soft non importa), il controllo passa ad una routine che "testa" la presenza, o meno, di una cartuccia (cartridge).

Nel riquadro sono riportate le routine Rom di Reset e di Test cartridge.

Questa routine cerca, a partire dalla locazione 32772 ($8004) il gruppo di caratteri "CBM80" e, se la trova, salta all'indirizzo specificato in $8000-$8001 per il Reset, e in $8002-$8003 per il Restore.

Detto ciò, appare chiaro che se nelle suindicate locazioni inseriremo i valori corrispondenti agli indirizzi di nostre routine l.m. potremo dirottare sia Reset che Restore.

Per il Restore (puntato dal terzo e quarto byte) il discorso è semplicissimo: basterà inserire l'indirizzo di "Warm start" ($FE69); oltre a sistemare alcune locazioni, questa routine permette infatti l'esecuzione del Restore SENZA ripristinare i vettori di interrupt con i loro valori originali; ecco perchè il nostro virus rimane attivo.

Nel caso del Reset le cose sono un po' più complesse e dovremo utilizzare un piccolo trucco.

Faremo uso del buffer di tastiera per far eseguire in modo diretto il comando "SYS 64767" che effettua un reset PARZIALE senza "danneggiare" l'interrupt; quindi salteremo a Warm start che si occuperà di "scaricare" il buffer, di ripristinare i colori e ridare il controllo al cursore.

Anche in questo caso il virus resterà attivo.

Il sistema descritto ha però un inconveniente: è necessario limitare la memoria disponibile a 32768 perdendo così 8 Kbyte di memoria. E' indispensabile infatti che la zona da $8000 a $8008 sia protetta da scrittura in modo da contenere i valori relativi ai nuovi indirizzi e ai codici di "CBM80".

Tutto ciò potrà sembrare di difficile comprensione (soprattutto ai principianti), ma è l'unica tecnica che permette di far coesistere Reset e interrupt.

IL VIRUS SI SCATENA

Anche per questa applicazione "virale" è utile riferirsi ad un esempio pratico.

Supponendo di aver realizzato un programma di una certa importanza, e volendolo mandare in visione a potenziali clienti, è opportuno prendere alcune precauzioni.

Potrebbe, in questo caso, risultare utile permettere all'utente di utilizza-

re il programma per sole tre volte perchè, alla quarta, apparirà un messaggio del genere:

ATTENZIONE. L'USO DI QUESTO PROGRAMMA E' ILLEGALE. CONTATTARE L'AUTORE PER L'AUTORIZZAZIONE PRIMA DI USARLO.

Sarà comunque data la possibilità di utilizzare il programma, ma a rischio e pericolo dell'utilizzatore.

Se questi è onesto, contatterà l'autore il quale, segnalandogli il pericolo del virus, lo metterà in guardia.

In caso contrario il virus sarà risvegliato e inizierà la sua azione distruttiva (e non dite che non eravate stati avvertiti!).

CONTEGGIO FATALE

Vediamo come è possibile realizzare un contatore che memorizzi il numero di utilizzazioni del programma.

Il demo proposto, come prima cosa, leggerà un particolare settore del disco (esattamente il settore 0 della traccia 1) alla ricerca di un numero indicante quante volte è stato utilizzato il programma.

Naturalmente la prima volta non troverà nulla e quindi scriverà nel settore il numero 1, proseguendo poi l'elaborazione del programma.

La seconda volta, trovato il numero, aggiornerà il contatore e proseguirà.

Giunto al limite (nel nostro caso, valore 2), stamperà il messaggio, aspetterà una risposta, e nel caso di uso illegale...

Vediamo ora come predisporre un dischetto atto a contenere sia il (o i) programmi, sia il contatore.

Per prima cosa è necessario utilizzare un dischetto vergine che, una volta formattato, conterrà il nostro software.

Per realizzare il contatore è opportuno utilizzare un file random che, non apparendo in directory, non insospettirà l'utente.

La scelta della traccia 1 e del settore 0 è stata fatta perchè è una delle zone del disco che vengono riempite per ultime e non sarà quindi neanche necessario controllare se è già allocata o meno perchè il disco, appena formattato, sarà senz'altro libero in quella zona.

Il fatto di non allocare il blocco ha inoltre il vantaggio di non "rubare" neanche uno del 664 blocchi, sempre per non insospettire l'utilizzatore.

Una volta formattato il dischetto, sarà sufficiente registrare il programma da mandare in visione, inglobando la routine virus nel corpo principale.

A questo punto il gioco è fatto.

UN VIRUS DISPETTOSO

Eccoci ora a parlare di questo secondo virus; quello della volta scorsa, infatti, avendo uno scopo didattico, non aveva alcuna conseguenza particolare; si limitava a manifestare la sua presenza.

Il nuovo virus, invece, sarà un po' più "pericoloso" in quanto, con frequenza casuale, sostituirà i caratteri presenti sullo schermo con altri, anch'essi casuali. Naturalmente il tempo di azione del virus è variabile e sarà quindi più difficile prevedere i suoi attacchi.

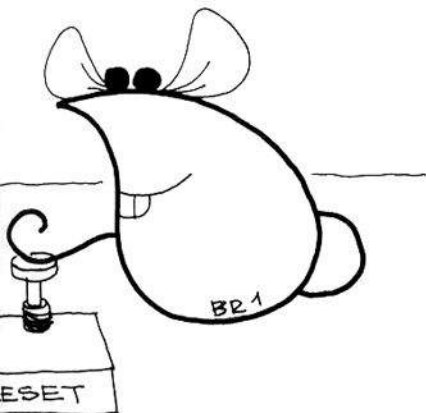
Come già detto, sia il Restore che il Reset saranno eseguiti, ma senza debellare il virus che alla prima occasione si ripresenterà.

Per prima cosa, dopo aver digitato e controllato il listato, registratelo su un dischetto appena formattato, dopo averlo copiato anche su un altro supporto magnetico (privo di dati importanti).

Per tre volte il programma funzionerà, ma alla quarta apparirà il fatidico messaggio; premete "U" e divertitevi ad osservare il virus in azione.

Certamente i suoi danni sono irrisori, in quanto nulla di ciò che appare sullo schermo è indispensabile, ma se il virus, invece di lavorare sullo schermo, avesse lavorato sul dischetto?

IL LISTATO DI ESEMPIO

Il lettore più attento si sarà accorto che il programma da proteggere (o, se preferite, da infettare) è quello presente al centro del listato di queste pagine (parte numerata da 100 a 190). Esso è un banale programma dimostrativo e si limita a far apparire, su video, i numerosi comandi disponibili sul C/64. E' ovvio che è vostro compito inserire il listato Basic cui applicare la procedura; altrettanto ovvio è che le righe precedenti (1 - 25) e successive (63971 - 63998) devono essere aggiunte al listato stesso.

Nulla di particolare da dire sul loader, è la solita sequenza di Read... Data, dotata di controllo di checksum e della linea Data corrente.

Qualche parola in più merita la parte relativa alla gestione del file random, argomento trattato di rado su questa rivista.

Nella linea 8 viene dapprima aperto il canale di colloquio con il drive (il 15) e, successivamente, anche quello (il n. 5) per il trattamento di un file relativo.

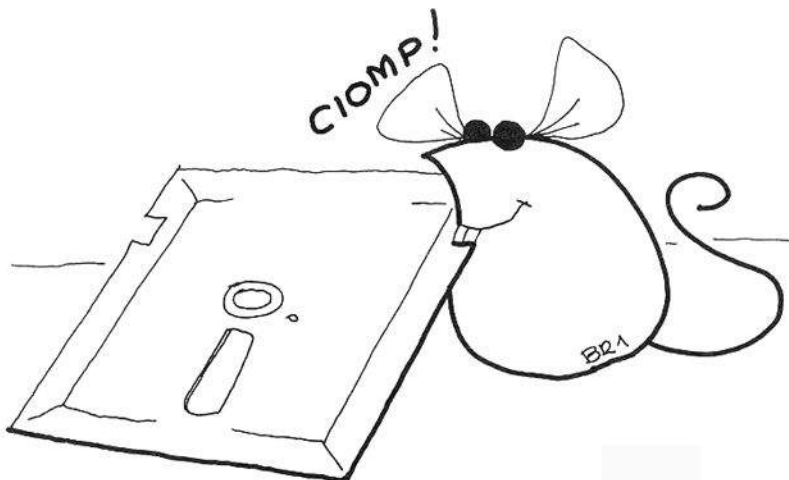
Nella stessa linea viene letto il contenuto, dal canale 5, della traccia 1, settore 0 del drive 0.

La linea successiva, la 9, fa puntare esattamente al sesto byte.

Viene quindi estratta una stringa della quale si calcola il valore che, se risulta superiore a 2, passa il controllo alla routine di avvertimento (14 - 19).

Se tutto va bene, cioè se il blocco non è ancora stato scritto oppure contiene un valore inferiore a 3, si incrementa il contatore e se ne riscrive il valore nello stesso settore (linee 10 - 11).

Nel caso venga tolto il dischetto mentre si effettua l'operazione, il controllo passa comunque alla routine di avvertimento.

Solo al termine delle procedure di controllo si può avere, finalmente, accesso al programma.

LE PAROLE DEL BASIC

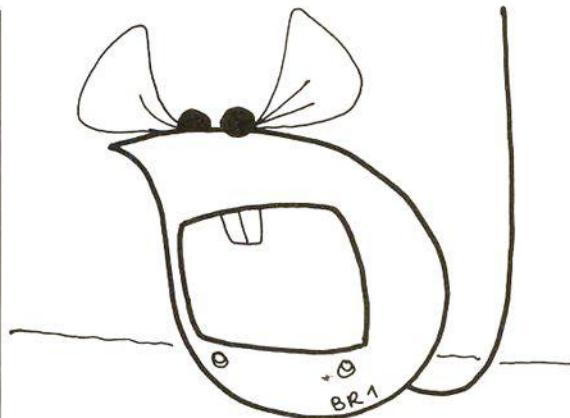
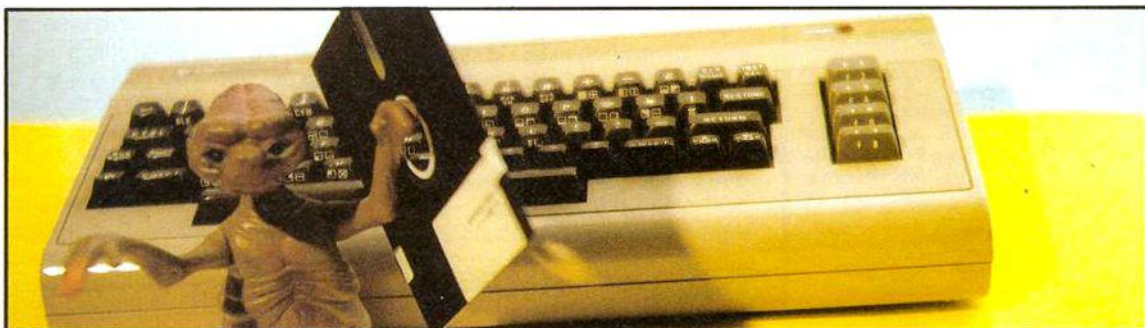
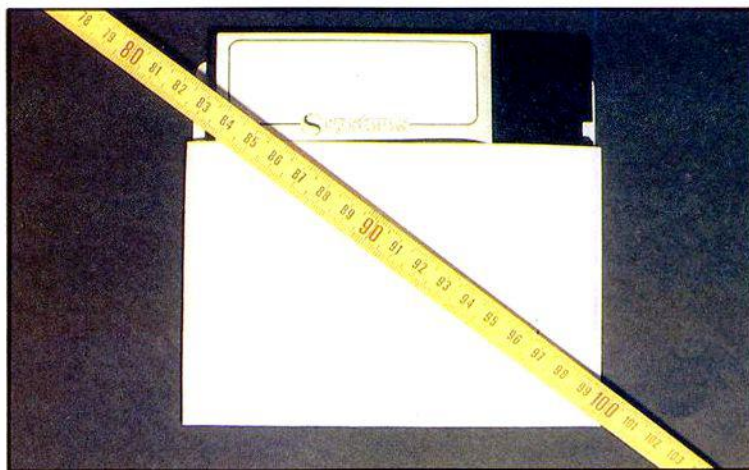
La Rom del Basic si estende da 40960 a 49151 ($A000 - $BFFF) e contiene, oltre alle parole chiave, anche i messaggi d'errore, vari puntatori e, naturalmente, le routine dell'interprete Basic.

Il demo proposto può dare spunto, ai più volenterosi, per un maggior approfondimento sull'argomento.

E' utile ricordare che ogni parola chiave (come ogni messaggio d'errore) è separata dalla successiva solo dal fatto che l'ultimo suo carattere ha il bit 7 acceso (ecco il motivo dell'AND 127 in linea 390 che lo "spegne").

Esiste, come è noto, la possibilità di trasferire la Rom Basic nella sottostante zona Ram in modo da modificare i comandi; questo argomento, molto interessante, è stato trattato sul n. 23 della rivista Personal Computer (articolo "Un Basic su misura") alla quale rimandiamo per ulteriore approfondimento.

```
1 REM CARICATORE VIRUS BY MICHELE MAGGI
2 PRINTCHR$(147):FOR I=0 TO 188:READ A:CK=CK+A
3 PRINTCHR$(19);PEEK(64)*256+PEEK(63):POKE32768+I,A:NEXT
4 IF CK=23194THEN 6
5 PRINT"ERRORE NEI DATA...":END
6 PRINTCHR$(147)"VIRUS V2.0 OK...":PRINT"PREMI UN TASTO PER IL DEMO!!!"
7 GETA$:IFA$=""THEN7
8 OPEN 15,8,15:OPEN 5,8,5,"#":PRINT#15,"B-R:5,0,1,0"
9 PRINT#15,"B-P:5,6":GET#5,A$:A=VAL(A$):IF A>2THEN13
10 A=A+1:PRINT#15,"B-P:5,5" :PRINT#5,A
11 PRINT#15,"B-W:5,0,1,0":INPUT#15,ER:IF ER<>0THEN13
12 CLOSE5:CLOSE15:GOTO 100:REM VIRUS INATTIVO
13 CLOSE5:CLOSE15
14 PRINTCHR$(147)
15 PRINT"ATTENZIONE!!!"
16 PRINT"L'USO DI QUESTO PROGRAMMA E' ILLEGALE"
17 PRINT"CONTATTARE L'AUTORE PER L'AUTORIZZAZIONE";
18 PRINT"PRIMA DI USARLO."
19 PRINT:PRINT:PRINT"PREMERE 'U' PER UTILIZZARE IL PROGRAMMA"
20 PRINT"OPPURE UN ALTRO TASTO PER FINIRE."
```

```
21 GET A$:IFA$="" THEN 21
22 IFA$="U" THEN 25
23 PRINT"ARRIVEDERCI...":END
24 :
25 SYS 32777:REM * VIRUS *
26 :
27 :
100 PRINTCHR$(147)"---------------------"
110 PRINT"PROGRAMMA DIMOSTRATIVO"
120 PRINT"---------------------"
130 PRINT:PRINT"STAMPA I COMANDI BASIC"
140 PRINT"LEGGENDOLI DALLA ROM":PRINT
150 FOR I=0 TO254
160 PRINTCHR$(PEEK(41118+I)AND 127);
170 IF PEEK(41118+I)>128 THEN PRINTCHR$(32)
180 NEXT
190 END
63970 :
63971 DATA 139,128,105,254,195,194,205
63972 DATA 056,048,120,169,030,141,020
63973 DATA 003,169,128,141,021,003,088
63974 DATA 169,255,133,055,169,127,133
63975 DATA 056,096,169,034,141,017,212
63976 DATA 141,016,212,141,015,212,141
63977 DATA 014,212,141,018,212,165,162
63978 DATA 205,138,128,176,080,162,000
63979 DATA 189,000,004,201,032,240,011
63980 DATA 205,137,128,208,006,173,027
63981 DATA 212,157,000,004,189,000,005
63982 DATA 201,032,240,011,205,137,128
63983 DATA 208,006,173,027,212,157,000
63984 DATA 005,189,000,006,201,032,240
63985 DATA 011,205,137,128,208,006,173
63986 DATA 027,212,157,000,006,189,232
63987 DATA 006,201,032,240,011,205,137
63988 DATA 128,208,006,173,027,212,157
63989 DATA 232,006,232,208,181,238,137
63990 DATA 128,076,049,234,001,010,169
63991 DATA 199,141,022,208,169,197,141
63992 DATA 022,208,162,255,032,179,238
63993 DATA 032,179,238,202,208,247,162
63994 DATA 009,189,177,128,157,118,002
63995 DATA 202,208,247,169,009,133,198
63996 DATA 076,105,254,083,089,083,054
63997 DATA 052,055,054,055,013,111,036
63998 END
```

```
*---------------------------*
* Routine di Reset 64738    *
*---------------------------*

FCE2 A2 FF    LDX #$FF
FCE4 78       SEI
FCE5 9A       TXS
FCE6 D8       CLD
FCE7 20 02 FD JSR $FD02
FCEA D0 03    BNE $FCEF
FCEC 6C 00 80 JMP ($8000)
:
:
*---------------------------*
* Controllo Cartridge       *
*---------------------------*

FD02 A2 05    LDX #$05
FD04 BD 0F FD LDA $FD0F,X
FD07 DD 03 80 CMP $8003,X
FD0A D0 03    BNE $FD0F
FD0C CA       DEX
FD0D D0 F5    BNE $FD04
FD0F 60       RTS
```

```
1  *--------------------------------*
2  * ESEMPIO DI INTRODUZIONE VIRUS  *
3  * DISTRUGGE A CASO I CARATTERI   *
4  * PRESENTI SULLO SCHERMO         *
5  * BY MICHELE MAGGI               *
6  * (C) 1988 BY SYSTEMS EDIT.      *
7  *--------------------------------*
8
9           HEX  888069FEC3  ;VALORI PER
10          HEX  C2CD3830    ;CARTRIDGE RESET
11
12 MAIN
13          SEI              ;DIROTTA
14          LDA  #<ENTRY     ;I VETTORI
15          STA  $0314       ;DI INTERRUPT
16          LDA  #>ENTRY     ;ALLA NUOVA
17          STA  $0315       ;ROUTINE
18          CLI              ;
19          LDA  #$FF        ;FISSA IL TOP
20          STA  55          ;DI MEMORIA
21          LDA  #$7F        ;A 32767
22          STA  56          ;
23          RTS              ;ED ESCE
24 ENTRY                     ;
25          LDA  #34         ;PREDISPONE
26          STA  54289       ;L'OSCILATORE
27          STA  54288       ;TRE PER LA
28          STA  54287       ;GENERAZIONE
29          STA  54286       ;DI NUMERI
30          STA  54290       ;RANDOM
31          LDA  162         ;OGNI VOLTA
32          CMP  BYTE        ;CHE 162 VALE PIU'
33          BCS  EXIT        ;DI BYTE
34          LDX  #0          ;ESEGUE IL
35 LOOP                      ;VIRUS E CERCA
36          LDA  $0400,X     ;UN CARATTERE
37          CMP  #$20        ;DIVERSO
38          BEQ  SKIP        ;DALLO SPAZIO
39          CMP  ASCII       ;ALL'INTERNO
40          BNE  SKIP        ;DELLO SCHERMO
41          LDA  54299       ;E LO SCAMBIA
42          STA  $0400,X     ;A CASO
43 SKIP                      ;
44          LDA  $0500,X     ;IDENTICA
45          CMP  #$20        ;PROCEDURA
46          BEQ  SKIP2       ;ANCHE PER LA
47          CMP  ASCII       ;SECONDA PARTE
48          BNE  SKIP2       ;DI SCHERMO
49          LDA  54299       ;
50          STA  $0500,X     ;
51 SKIP2                     ;
52          LDA  $0600,X     ;IDENTICA
53          CMP  #$20        ;PROCEDURA
54          BEQ  SKIP3       ;ANCHE PER LA
55          CMP  ASCII       ;TERZA PARTE
56          BNE  SKIP3       ;DI SCHERMO
57          LDA  54299       ;
58          STA  $0600,X     ;
59 SKIP3
60          LDA  $06E8,X     ;IDENTICA
61          CMP  #$20        ;PROCEDURA
62          BEQ  SKIP4       ;ANCHE PER LA
63          CMP  ASCII       ;QUARTA PARTE
64          BNE  SKIP4       ;DI SCHERMO
65          LDA  54299       ;
66          STA  $06E8,X     ;
67 SKIP4
68          INX              ;
69          BNE  LOOP        ;
70          INC  ASCII       ;INCREMENTA IL
71 EXIT                      ;CARATTERE DI
72          JMP  $EA31       ;CONFRONTO ED ESCE
73 ASCII    DFB  1           ;
74 BYTE     DFB  10          ;TIMER
75
76 *--------------------------------*
77 * ROUTINE DI SIMULAZIONE RESET   *
78 * RIPORTA I COLORI AI VALORI     *
79 * DI DEFAULT, INTERROMPE IL      *
80 * PROGRAMMA IN CORSO MA NON      *
81 * ELIMINA IL VIRUS               *
82 *--------------------------------*
83
84 RESET
85          LDA  #199        ;SIMULA IL
86          STA  53270       ;RESET
87          LDA  #197        ;
88          STA  53270       ;
89          LDX  #255        ;USA UNA ROUTINE
90 DELAY                     ;DEL S.O.
91          JSR  $EEB3       ;PER GENERARE
92          JSR  $EEB3       ;UN RITARDO
93          DEX              ;DI QUALCHE
94          BNE  DELAY       ;MILLISECONDO
95          LDX  #9          ;CARICA IL
96 WRITE                     ;MESSAGGIO
97          LDA  TESTO-1,X   ;CON LA SYS
98          STA  630,X       ;E LO METTE
99          DEX              ;NEL BUFFER
100         BNE  WRITE       ;DI TASTIERA
101         LDA  #$09        ;
102         STA  198         ;
103         JMP  $FE69       ;WARM START
104 TESTO
105         TXT  'sys64767'
106         HEX  0D
```

MISURIAMO LE PAROLE

Un file sequenziale, si sa, è un lungo fiume di parole che potrebbe non andare mai "a capo". Con opportuno software, però...

di **Alessandro de Simone**

Vi è mai capitato di dover visualizzare, su schermo o stampante, il contenuto di un file generato da un word processor?

Di solito, se non si utilizza il word processor che lo ha generato, i caratteri alfanumerici appaiono l'uno di seguito all'altro e le parole che giungono alla fine del rigo (di video o di carta) vengono spezzate in modo anomalo, con conseguente difficoltà di lettura ("anestetismi" a parte).

I due programmi che proponiamo (e che girano su un qualsiasi "piccolo" computer Commodore) sono basati su un'unica routine incaricata di esaminare, byte per byte, i caratteri appartenenti al file sequenziale (registrato su nastro o disco). Mentre il primo, però, suddivide una frase in modo "intelligente", su video o stampante, il secondo gira solo disponendo di una stampante Mps-803 compatibile.

LA ROUTINE DI BASE

Come ci è capitato spesso di ricordare, un file sequenziale, di solito, è costituito da un'unica sequenza di caratteri alfanumerici, interrotta, eventualmente, da un carattere che possiamo considerare un po' particolare, il ritorno carrello, Chr$(13), volgarmente detto carattere di "a capo".

Visualizzando il file, accade che i suoi caratteri vengono stampati l'uno di seguito all'altro, senza interruzione, tranne nel caso in cui si incontri un Chr$(13); oppure, ovviamente, nel caso in cui il dispositivo fisico (video o stampante) non sia più in grado di contenere il rigo stesso.

In quest'ultimo caso, infatti, la visualizzazione continua al rigo successivo, non rispettando, però, le regole sintattiche della suddivisione in sillabe nè, tantomeno, allineamenti particolari (le cosiddette giustificazioni e centrature).

Supponendo, quindi, di avere un file sequenziale lungo un migliaio di byte, costituito da parole, spazi e ritorni carrello, il problema è quello di riportare su video (o stampante) un certo numero di parole "intere". Nel caso in cui, giunti in prossimità della fine del rigo, una parola non possa essere ospitata per intero, questa deve essere scritta all'inizio del rigo successivo. Naturalmente deve essere offerta la possibiltà di scegliere

SCHEDA TECNICA

Software di utilità da applicare ai file sequenziali.Hardware richiesto: C/64, C/16, Plus/4, C/128, Vic-20.Non facilmente adattabile ad altri computer.Il solo listato n. 2 richiede stampante MPS-803 o compatibili.Richiede drive o registratore.Consigliato ai lettori più intraprendenti.

Anche i programmi pubblicati in queste pagine sono contenuti nel disco "Directory" di questo mese.

DA UN'IDEA ALL'ALTRA

I programmi presentati in queste pagine sono un modesto e limitato tentativo di indicare, ai lettori più volenterosi, la via da seguire per incrementare la propria esperienza.

Da un vecchio numero di C.C.C, infatti, è stato riesumato un programma che, pur ridotto all'osso, presentava caratteristiche piuttosto interessanti per chi ha spesso a che fare con file provenienti, tipicamente, da word processor.

La richiesta, da parte di numerosi lettori, di sfruttare la capacità di riprogrammazione dei caratteri offerta delle stampanti, imponeva, d'altra parte, che si affrontasse l'argomento.

Ed ecco, quindi, che al dilettevole si aggiunge l'utile. Non solo, ma si indicano alcune procedure per snellire il lavoro di riprogrammazione dei caratteri e si suggerisce l'introduzione di altre opzioni, tutte di notevole interesse.

Tale modo di agire, quindi, è del tutto coerente con la "linea" da sempre suggerita da Commodore Computer Club: imparare a programmare è facile e, ancora più semplice, è quell'attività che consiste nello studiare con attenzione i programmi già sviluppati da altri ed apportare le modifiche che più riteniamo utili per le nostre esigenze.

E così, modifica dopo modifica, si perviene spesso a programmi di notevole pregio, di innegabile utilità e che, spesso, siamo portati ad utilizzare più frequentemente di altri programmi, cosiddetti "professionali".

la lunghezza massima del rigo (misurato in numero di caratteri) e bisogna prevedere una serie di situazioni particolari che possono verificarsi.

COME FUNZIONA

Supponiamo ancora, per semplificare la spiegazione, che il file in questione sia costituito da alcune centinaia di parole, tutte eguali tra loro, e che il termine in questione sia "Vocabolo", lungo 8 caratteri. Naturalmente, tra una parola e l'altra, sarà presente un solo spazio.

Se fissiamo in 33 la lunghezza massima di rigo che desideriamo visualizzare, ognuno di questi potrà contenere solo 3 parole "Vocabolo"; tenendo presente la lunghezza di ciascun termine, e lo spazio tra due termini successivi, risulta...

8 + 1 + 8 + 1 + 8 = 26

Se, infatti, si volesse aggiungere, sullo stesso rigo, un altro "Vocabolo", risulterebbe...

8 + 1 + 8 + 1 + 8 + 1 +8 = 35

...che è un valore superiore a quello massimo impostato (33).

Dovrebbe ora esser chiaro quale sia il problema che il programma deve risolvere.

IL PRIMO LISTATO

Il primo programma che compare in queste pagine è in grado di leggere un file sequenziale da disco e di riversarlo su video o stampante secondo le direttive appena descritte.

Coloro che posseggono il solo registratore a cassette (comprate un drive, ragazzi!) devono limitarsi a modificare la riga 190 digitando soltanto "Open 1" oppure ricorrendo ad una sintassi più sofisticata in grado di selezionare un file dotato di nome (per tale scopo si veda il libretto di istruzioni del computer, paragrafo "Gestione dei File").

Ma vediamo ora quali problemi possono presentarsi e, soprattutto, in che modo vengono risolti.

Dopo aver aperto, in lettura, il file da leggere, viene aperto, in scrittura, il file da riversare sulla periferica precedentemente indicata (180 - 190). Tenendo conto che la lunghezza massima del rigo è stata associata alla variabile LU (riga 150) si tratta, ora, di fare in modo da non superarla.

Il programma, pertanto, inizia definendo una stringa nulla (Y$) destinata a contenere il rigo che verrà in seguito stampato; quindi incomincia a leggere il file sequenziale, un byte alla volta, che deposita nella variabile X$. Non appena viene incontrato uno spazio (codice Ascii 32, riga 250, simbolo implicito di "fine parola intera") si controlla che la lunghezza della stringa X$, sommata a quella di Y$, non sia maggiore di LU. Se la somma è minore di LU, ad Y$ viene aggiunta, in coda, la stringa X$; quest'ultima viene quindi annullata, e si riprende la lettura del file.

Così si procede, parola dopo parola, finchè la lunghezza di Y$ rimane inferiore a LU. Non appena si verifica il contrario, Y$ viene stampata, subito dopo viene annullata e al suo posto viene allocata la stringa X$ che rappresenta l'ultima parola, letta dal file, che non può essere stampata alla fine del rigo precedente.

L'eventuale presenza di un Chr$(13) viene interpretata come un "a capo" forzato ed il rigo "accumulato" fino a quel momento viene subito stampato, indipendentemente dalla sua lunghezza.

Si può verificare, però, un caso particolare, vale a dire il caso in cui non si incontrano caratteri di spazio oppure (è lo stesso) una parola, da sola, sia più lunga di LU.

In questo caso, ovviamente, non c'è altro da fare se non stampare un numero di caratteri eguale a LU, andare a capo e continuare nella procedura.

Tutte le funzioni descritte vengono svolte dal gruppo di "filtri" If...Then

che suggeriamo di studiare attentamente. I lettori più volenterosi potrebbero estrarne il diagramma di flusso e tentare di ottimizzarlo. I lettori super, poi, potrebbero inserire una subroutine di suddivisione in sillabe per raggiungere il Top della sofisticazione.

IL SECONDO LISTATO

Il secondo listato non fa altro che sfruttare la procedura già esaminata.

La differenza consiste nel fatto che funziona solo con la stampante (una qualsiasi 801 oppure 803-compatibile) dal momento che è stata aggiunta una routine che, riprogrammando i caratteri della periferica, consente di riprodurre caratteri più piccoli. La conseguenza immediata è la possibilità di stampare righe contenenti oltre 100 caratteri invece dei soliti 80.

La quantità massima di caratteri stampabili dipende sia dal modello di stampante usato sia dal "tipo" di caratteri che costituiranno il rigo stesso. Per ottimizzare la personalizzazione, infatti, non tutti i caratteri appartengono ad una griglia standard: quelli più "grandi" (come gli alfabetici) sono contenuti in una matrice di 3 x 5, mentre i più piccoli (numero 1, simboli di puntaggiatura eccetera) in una più modesta matrice di 1 x 5.

Si utilizza, insomma, una sorta di stampa proporzionale che il lettore, volendo, può miniaturizzare ulteriormente.

Per semplicità, ma soprattutto per evitare lunghe digitazioni di Data, vengono ri-programmati solo i caratteri alfanumerici ed i simboli di punteggiatura (codici Ascii da 32 a 96); gli altri sono stati volutamente trascurati ma, lo ripetiamo, possono essere aggiunti con una certa facilità dai lettori più disinvolti con il Basic.

La routine di ridefinizione dei caratteri è stata presa, pari pari, da un programma già pubblicato su C.C.C. (n. 15, "Stampa compattata") ed il suo funzionamento lo accenniamo

soltanto, dal momento che il lettore, studiando il libretto di istruzioni della stampante, può meglio comprenderne la tecnica.

In un vettore stringa WO$ (già predisposto per 128 caratteri, ma utilizzato per un numero inferiore) vengono allocate le stringhe, una per ciascun carattere, contenenti i codici Ascii dei caratteri da programmare. Tali codici sono, ovviamente, trascritti nei Data (righe da 580 in poi) secondo un codice ben definito. Ogni carattere, infatti, è rappresentato dai valori contenuti tra due zeri successivi; il carattere spazio (che è il primo ad esser ridefinito) è quindi rappresentato dai tre 128 (riga 580).

Il punto esclamativo (codice 33, successivo allo spazio che ha codice 32) sarà rappresentato dal solo valore 23; il carattere di virgolette ("), di codice 34, sarà individuato dai valori 3, 128, 3, e così via, carattere dopo carattere, tranne l'ultimo Data che, essendo negativo, ha il solo compito di comunicare alla routine di inizializzazione di aver letto tutti i Data presenti nel programma.

Un'ultima nota per ciò che riguarda la lettura del file: siccome i caratteri riprogrammati sono solo quelli alfanumerici maiuscoli, è stato necessario introdurre un filtro (riga 310) che ha il compito di trasformare, in maiuscolo, eventuali caratteri registrati in minuscolo.

SFIDA AI LETTORI

Il programma si presta per sofisticazioni di notevole interesse. Suggeriamo, pertanto, di aggiungere varie utility:

- Programmare tutti i caratteri dello standard Commodore.
- Inserire l'opzione "Stampa normale / riprogrammata".
- Inserire l'opzione "Video / Stampante".
- Offrire la possibilità di suddivisione in sillabe, servendosi, magari, del listato specifico che compare in queste stesse pagine.

```
 APPLICAZIONI
 MISURIAMO LE PAROLE
 UN FILE SEQUENZIALE, SI SA, E' UN LUNGO FIUME DI PAROLE
CHE POTREBBE NON ANDARE MAI "A CAPO". CON OPPORTUNO
SOFTWARE, PERO'...
 ALESSANDRO DE SIMONE
 VI E' MAI CAPITATO DI DOVER VISUALIZZARE, SU SCHERMO O
STAMPANTE, IL CONTENUTO DI UN FILE GENERATO DA UN WORD
PROCESSOR?
 #DI SOLITO, SE NON SI UTILIZZA IL WORD PROCESSOR CHE LO
HA GENERATO, I CARATTERI ALFANUMERICI APPAIONO L'UNO DI
SEGUITO ALL'ALTRO E LE PAROLE CHE GIUNGONO ALLA FINE DEL
RIGO (DI VIDEO O DI CARTA) VENGONO SPEZZATE IN MODO
ANOMALO, CON CONSEGUENTE DIFFICOLTA' DI LETTURA
("ANESTETISMI" A PARTE).
 #I DUE PROGRAMMI CHE PROPONIAMO (E CHE GIRANO SU UN
QUALSIASI "PICCOLO" COMPUTER COMMODORE) SONO BASATI SU
UN'UNICA ROUTINE INCARICATA DI ESAMINARE, BYTE PER BYTE, I
CARATTERI APPARTENENTI AL FILE SEQUENZIALE (REGISTRATO SU
NASTRO O DISCO). MENTRE IL PRIMO, PERO', SUDDIVIDE UNA
FRASE IN MODO "INTELLIGENTE", SU VIDEO O STAMPANTE, IL
SECONDO GIRA SOLO DISPONENDO DI UNA STAMPANTE MPS-803
COMPATIBILE.

 LA ROUTINE DI BASE
 #COME CI E' CAPITATO SPESSO DI RICORDARE, UN FILE
SEQUENZIALE, DI SOLITO, E' COSTITUITO DA UN'UNICA SEQUENZA
DI CARATTERI ALFANUMERICI, INTERROTTA, EVENTUALMENTE, DA
UN CARATTERE CHE POSSIAMO CONSIDERARE UN PO' PARTICOLARE,
IL RITORNO CARRELLO, CHR$(13), VOLGARMENTE DETTO CARATTERE
DI "A CAPO".
 #VISUALIZZANDO IL FILE, ACCADE CHE I SUOI CARATTERI
VENGONO STAMPATI L'UNO DI SEGUITO ALL'ALTRO, SENZA
INTERRUZIONE, TRANNE NEL CASO IN CUI SI INCONTRI UN
CHR$(13); OPPURE, OVVIAMENTE, NEL CASO IN CUI IL
DISPOSITIVO FISICO (VIDEO O STAMPANTE) NON SIA PIU' IN
```

Esempio di stampa compattata, riprodotta dal secondo programma di queste pagine. Si noti l'impostazione dei soli caratteri maiuscoli, la variabilità della loro "matrice" e, soprattutto, l'estrema compattezza dei caratteri. Al termine di un rigo, inoltre, non compaiono mai parole riprodotte in modo parziale.

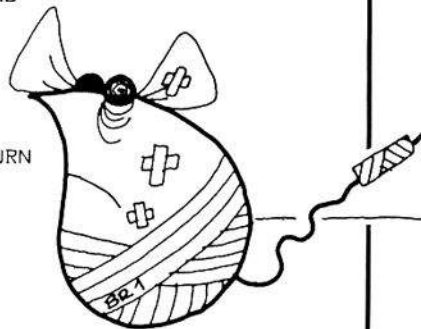
• Memorizzare l'intero file in un vettore (ciascun rigo stampabile associato al corrispondente elemento della matrice, in successione) e, alla fine, riportare su stampante una sola, due oppre tre colonne di testo affiancate.

Soprattutto quest'ultima opzione potrebbe trasformare il listato proposto in un efficace ed utilissimo formattatore di testi.

I lettori più libidinosi, inoltre, potranno divertirsi ad inserire una routine di conversione, per adattare, al codice Ascii Commodore, i file provenienti da word processor che memorizzano i testi secondo codici poco standardizzati.

I pazzi furiosi, infine, possono introdurre una routine che individui i codici particolari di formattazione testo (giustificazione, centratura, sottolineatura e così via) e li renda operanti anche dopo la conversione.

Se siete in grado di raggiungere tali livelli di follia, però, non ditelo a nessuno; potrebbero, giustamente, preoccuparsi del vostro stato mentale.

```
100 PRINT CHR$(147) "QUESTO PROGRAMMA CONSENTE DI LEGGERE ";
110 PRINT "UN FILE E DI RIPORTARLO, SU VIDEO ";
120 PRINT "O STAMPANTE, IN RIGHE DI LUNGHEZZA IMPOSTATA.":PRINT:PRINT
130 PRINT: PRINT: PRINT CHR$(18) "PREMI UN TASTO"
140 GET A$: IF A$="" THEN 140
150 PRINT: INPUT"LUNGHEZZA MASSIMA";LU: LU=LU-1
160 IF LU<2 THEN 150
170 INPUT "FILE";NF$
180 DN=4: INPUT"VIDEO / STAMPANTE";DN$: IF DN$<>"S" THEN DN=3
190 OPEN 1,8,8,NF$: OPEN 3,DN
200 Y$="": REM BY A. DE SIMONE
210 X$="": REM PER COMMODORE COMPUTER CLUB
220 GET#1,Q$: IF ST>0 THEN CLOSE1: CLOSE3: END
230 GET F$: IF F$<>"" THEN PRINT#3: CLOSE3: CLOSE1: END
240 A=ASC(Q$)
250 IF A=32 THEN GOSUB290: GOTO210
260 IF A=13 THEN GOSUB340: GOTO200
270 IF LEN(X$)+1>LU THEN GOSUB320: GOTO210
280 X$=X$+Q$: GOTO220
290 IF LEN(X$)+1>LU THEN PRINT#3,Y$: Y$=X$: RETURN
300 IF LEN(Y$)+LEN(X$)>LU THEN PRINT#3,Y$: Y$=X$: RETURN
310 Y$=Y$+Q$+X$: LY=LEN(Y$): RETURN
320 IFLY>0 THEN PRINT#3,Y$: Y$=X$+Q$: RETURN
330 PRINT#3,X$: X$="": Y$="": LY=0: RETURN
340 IF(LEN(X$)>LU)OR(LEN(X$)+LEN(Y$))>LU THEN 360
350 Y$=Y$+" "+X$: PRINT#3,Y$: RETURN
360 PRINT#3,Y$: PRINT#3,X$: LY=0: RETURN
370 END
```

```
100 REM LETTURA FILE E SCRITTURA SU STAMPANTE
110 REM BY A. DE SIMONE E F. SORGATO
120 PRINT CHR$(147)"QUESTO PROGRAMMA CONSENTE DI LEGGERE"
130 PRINT"UN FILE SEQUANZIALE DA DISCO E DI"
140 PRINT"RIVERSARLO SU STAMPANTE MODIFICANDONE"
150 PRINT"IL CARATTERE (PIU' PICCOLO).":PRINT
160 PRINT"LA LUNGHEZZA MASSIMA DEL RIGO STAMPABILE";
170 PRINT"DIPENDE DALLE ";
180 PRINT"CARATTERISTICHE DELLA STAMPANTE STESSA"
190 PRINT: PRINT CHR$(18)"PREMI UN TASTO"
200 GET A$:IF A$="" THEN 200
210 GOSUB 450: REM INIZIALIZZA MATRICE
220 INPUT "LUNGHEZZA MASSIMA";LU: LU=LU-1
230 INPUT "INTERLINEA (0/1)";LN
240 INPUT "FILE";NF$
250 OPEN 2,8,8,NF$: OPEN 1,4
260 Y$="": LY=0
270 X$=""
280 GET#2,Q$: IF ST>0 THEN CLOSE2: CLOSE1: END
290 GET F$: IF F$<>"" THEN PRINT#1: CLOSE1: CLOSE2: END
300 A=ASC(Q$)
310 IF A>128 THEN Q$=CHR$(A-128)
320 IF A= 32 THEN GOSUB360: GOTO270
330 IF A= 13 THEN GOSUB410: GOTO260
340 IF LEN(X$)+1>LU THEN GOSUB390: GOTO270
350 X$=X$+Q$: GOTO280
360 IF LEN(X$)+1>LU THEN WO$=Y$: GOSUB520: Y$=X$: RETURN
370 IF LEN(Y$)+LEN(X$)>LU THEN WO$=Y$: GOSUB390: Y$=X$: RETURN
380 Y$=Y$+Q$+X$: LY=LEN(Y$): RETURN
390 IFLY>0 THEN WO$=Y$: GOSUB520: Y$=X$+Q$: RETURN
400 WO$=X$: GOSUB520: X$="": Y$="": LY=0: RETURN
410 IF(LEN(X$)>LU) OR (LEN(X$)+LEN(Y$))> LU THEN 430
420 Y$=Y$+" "+X$: WO$=Y$: GOSUB520: RETURN
430 WO$=Y$: GOSUB520: WO$=X$: GOSUB520: LY=0: RETURN
440 END
450 DIM WO$(128): T=32 :PRINTCHR$(147)".....(DEVO CONTARE FINO A 96)"
460 READ A: IF A<128 THEN J=128
470 WO$(T)=WO$(T)+CHR$(A+J): IF A=0 THEN T=T+1
480 J=0: IF A<0 THEN RETURN
490 PRINT CHR$(19)T: GOTO 460
500 :
510 REM SUBROUTINE DI STAMPA
520 PRINT#1,CHR$(8);
530 FOR K=1 TO LEN(WO$): PRINT#1,WO$(ASC(MID$(WO$,K,1)));: NEXT
540 PRINT#1,CHR$(13);
550 IF LN<>0 THEN PRINT#1,CHR$(13);
560 WO$="": RETURN
570 :
580 DATA 128,128,128,0,23,0,3,128,3,0
590 DATA 10,31,10,31,10,0,18,21,31,21,9,0,17,8,4,2
600 DATA 17,0,26,21,26,20,0,2,1,0,14,17,0
610 DATA 17,14,0,21,14,31,14,21,0,4,4,31,4,4,0
620 DATA 16,12,0,4,4,4,4,0,16,0,24,4,3,0
630 DATA 31,17,31,0,31,0,29,21,23,0,17,21,31,0
640 DATA 7,4,31,0,23,21,13,0,31,21,29,0
650 DATA 1,1,31,0,31,21,31,0,23,21,31,0
660 DATA 18,0,16,10,0,4,10,17,17,0,10,10
670 DATA 10,10,0,17,17,10,4,0,2,1,21,5,2,0
680 DATA 31,17,21,23,0,31,5,31,0,17,31,21
```

```
690 DATA 31,0,31,17,17,0,31,17,27,14,0
700 DATA 31,21,17,0,31,5,1,0,31,17,29,0
710 DATA 31,4,31,0,17,31,17,0,17,31,1,0,31,6,25
720 DATA 0,31,16,16,0,31,2,4,2,31,0
730 DATA 31,2,4,31,0,31,17,31,0,31,5,7,0
740 DATA 31,17,31,16,0,31,13,23,0,23,21,29,0
750 DATA 1,31,1,0,31,16,31,0,15,24,15,0
760 DATA 31,8,4,8,31,0,27,4,27,0
770 DATA 3,30,3,0,25,21,19,0,31,17,0,30
780 DATA 21,21,18,0,17,31,0
790 DATA 2,31,2,0,8,28,8,8,8,0
800 DATA -1
810 END
```

```
DIVISIONE IN SILLABE

  100 REM  ESEMPIO D'USO
  110 REM  SCOMPOSIZIONE IN SILLABE
  115 REM  PER QUALSIASI COMPUTER
  120 REM  (TRATTO DA C.C.C N. 27)
  121 :
  122 REM  RUBRICA "ENCICLOPEDIA DI ROUTINE BA
      SIC"
  123 :
  130 INPUT "VOCABOLO";Y$
  140 GOSUB 11600:FOR I=0 TO 10:IF Y$(I)="" TH
      EN I=10
  145 PRINT"SILLABA"I;CHR$(18)Y$(I):NEXT:GOTO
      130
  150 :
 9999 END
11600 X1=0:FOR X9=0 TO 10:Y$(X9)="":NEXT:X9=0
11605 X1=X1+1:IF X1>LEN(Y$) OR X9>10 THEN RETU
      RN
11610 FOR X0=0 TO 2:X$(X0)=MID$(Y$,X1+X0,1):X$
      =X$(X0)
11615 X(X0)=X$="I" OR X$="U" OR X$="O" OR X$="
      A" OR X$="E":NEXT:IF X(0)=0 THEN 11675
11625 IF X(1)=0 THEN 11655
11635 IF X$(0)="I" OR X$(0)="U" THEN 11675
11640 GOTO 11680
11650 X3=(X(2)=0)+2:ON X3GOTO 10075,11680
11655 IF X(2)=-1 THEN 11680
11660 IF X$(1)=X$(2) THEN 11670
11665 IF X$(1)="S" OR X$(1)="G" OR X$(2)="R" O
      R X$(2)="L" OR X$(2)="H" THEN 11680
11670 Y$(X9)=Y$(X9)+X$(0)+X$(1):X1=X1+1:X9=X9+
      1:GOTO 11605
11675 Y$(X9)=Y$(X9)+X$(0):GOTO 11605
11680 Y$(X9)=Y$(X9)+X$(0):X9=X9+1:GOTO 11605
11690 REM  X$(), Y$() X$,X0,X1,X3,X9
11699 REM  SCOMPOSIZIONE IN SILLABE
```

CAMPUS

LA TUA UNIVERSITA' DEL SOFTWARE

LABORATORIO SOFTWARE DI COMMODORE COMPUTER CLUB

ESORDIENTI

- Le popolazioni informatiche

ESPERTI

- Viaggio al centro del Basic
- Un antilist a prova di hacker

CAMPIONI

- Tutti i tempi del video
- Il nome del Kernal

DIRECTORY

Elenco dei floppy disk "Directory", finora usciti, e loro corrispondenza con il numero del fascicolo di Commodore Computer Club (C.C.C.):

1 :C.C.C. n. 34
2 :C.C.C. n. 35 + n. 36
3 :C.C.C. n. 37
4 :C.C.C. n. 38
5 :C.C.C. n. 39
6 :C.C.C. n. 40
7 :C.C.C. n. 41
8 :C.C.C. n. 42
9 :C.C.C. n. 43
10 :C.C.C. n. 44
11 :C.C.C. n. 45
12 :C.C.C. n. 46
13 :C.C.C. n. 47
14 :C.C.C. n. 48
15 :C.C.C. n. 49
16/17 :C.C.C. n. 50 + C.C.C. n. 51
18 :C.C.C. n. 52
19 :C.C.C. n. 53
20 :C.C.C. n. 54 + C.C.C. n. 55
21 :C.C.C. n. 56 + C.C.C. n. 57
22 :C.C.C. n. 58

COME PROCURARSI "DIRECTORY"

Avvertiamo i lettori che NON ci è assolutamente possibile inviare i programmi su nastro, per intuibili motivi di economia ed affidabilità del nastro cassetta.
"Directory" può quindi esser richiesto solo su disco, inviando L. 12000 per ciascun floppy desiderato, oltre alle spese di confezione, imballo e spedizione (L. 3000 per spedizione "normale"; L. 6000 per inoltro a mezzo raccomandata)

Si sottolinea che NON ci è possibile inviare materiale contrassegno.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo di indirizzo (ed eventuale telefono), ma anche il titolo del prodotto desiderato; esempio:

Mario Rossi
Via dei giardini 12
00123 Roma
(Tel. 06/12.34.56)

Desidero ricevere:

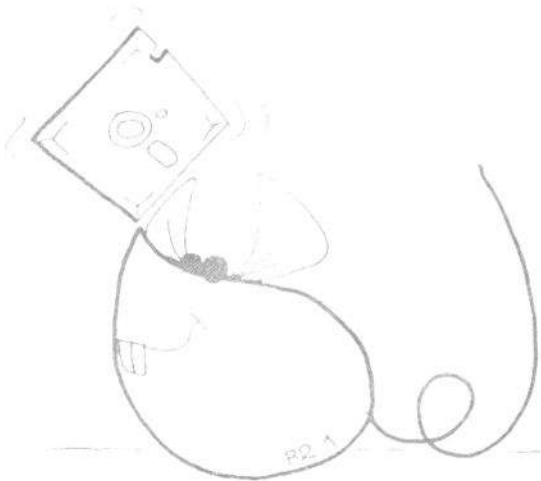
"Directory" N.1, N.7, N.12

La somma risulta:

Dischetti L. 12000 x 3 = L. 36000
Spediz. raccom. L. 6000
Totale: L. 42000

E' intuitivo che, con lo stesso ordinativo, potete chiedere altri prodotti della Systems Editoriale (arretrati, libri, dischi e così via) il cui prezzo di vendita è indicato nelle apposite pagine informative presenti anche in questo stesso fascicolo.

Si consiglia, per coloro che dispongono di un conto corrente presso QUALSIASI banca o Istituto di Credito, di inviare la somma dovuta non a mezzo c/c postale, ma a mezzo assegno (non trasferibile o sbarrato due volte) da inviare, in busta chiusa, al nostro indirizzo, corredandolo di lettera di accompagnamento.

In questo modo la spedizione dei prodotti richiesti sarà evasa in tempi notevolmente più brevi.

LE POPOLAZIONI INFORMATICHE

Uno degli argomenti tipicamente "computerizzabili" è la statistica, ma è noioso introdurre dati su cui operare. Vediamo di aggirare il problema; sia in Basic sia in Turbo - Pascal

di **Alessandro de Simone**

DAI POLLI AL COMPUTER

Tranquilli, non inizieremo con il solito problema dei polli secondo cui se io mangio un pollo intero, e tu rimani digiuno, in media abbiamo mangiato mezzo pollo a testa.

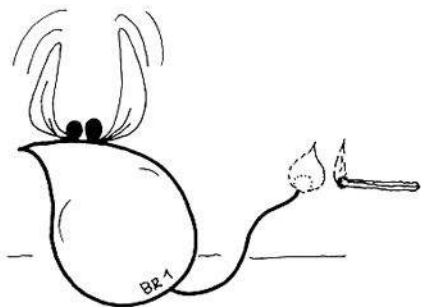
La statistica è una cosa seria e, se applicata ai "grandi" numeri, è certamente in grado di evidenziare dati altrimenti difficili da rilevare in altro modo.

Il primo argomento che, di solito, si incontra studiando la statistica è quello relativo alla determinazione della media aritmetica che, portata (erroneamente) alle estreme conseguenze, può generare anche l'assurdità del mezzo pollo a testa.

Non bisogna dimenticare, infatti, che ad ogni problema corrisponde un ben definito metodo risolutivo (altrimenti non ve ne sarebbero tanti, vi pare?) e, soprattutto, che certi ragionamenti possono essere seguiti, PER DEFINIZIONE, solo se entrano in gioco numerosi dati, per di più omogenei tra loro.

Tanto per tornare sull'esempio di prima (sostenuto fin dalla notte dei tempi dai soliti scettici), le due categorie di persone (io che mangio e tu che sei digiuno) non sono omogenee e, di conseguenza, non possono essere paragonate tra loro nè, tantomeno, sottoposte ad elaborazioni che partono da ben altri presupposti.

Può aver senso determinare il valor medio del numero di polli consumati in un ristorante nell'arco di un certo periodo di tempo (giorno, settimana, mese, oppure solo a pranzo o solo a cena e così via) oppure confrontare tra loro le percentuali di secondi piatti ordinati (pollo 32%, pesce 12%, bistecca 56%); gli avventori che entrano in quel determinato ristorante possono, infatti, essere considerati sufficientemente omogenei, cioè gente affamata, presumibilmente appartenente ad una categoria sociale ben definita, e ad una precisa tradizione culinaria.

by Marco Mietta

```
100 REM INTRODUZIONE ALLA STATISTICA
110 REM STUDI SULLA "POPOLAZIONE"
120 :
130 Y$=CHR$(147): PRINT Y$
140 PRINT"QUESTO PROGRAMMA DETERMINA IL"
150 PRINT"VALOR MEDIO DEL CONTENUTO DELLA"
160 PRINT"MEMORIA DEL TUO COMPUTER":PRINT
170 INPUT"DALLA LOCAZIONE";LI: REM DETERMINAZIONE
180 INPUT "ALLA LOCAZIONE";LF: REM INTERVALLO
190 TT = LF-LI+1:             REM NUMERO LOCAZIONI
200 PRINT Y$ "....... DEVO ARRIVARE A:" LF
210 X=0: FOR I=LI TO LF:      REM ESAME DELLE LOC.
220 X = X + PEEK(I)
230 PRINT CHR$(19) I: NEXT
240 PRINT Y$ "LA MEDIA VALE:" X/TT
250 PRINT "LE LOCAZIONI ESAMINATE SONO:" TT
260 PRINT"NUMERATE DA:" LI "A:" LF
270 END
```

Le 65536 locazioni di memoria del vostro computer possono rappresentare una "popolazione" statistica sufficientemente valida

Se, in un certo periodo, si dovessero notare valori anomali, diversi dal consueto, ciò potrà aiutare nel capire o individuare determinati fenomeni. Se, ad esempio, si dovesse notare un aumento del consumo dei polli, di solito più economici delle bistecche, si potrà ricercare la causa dell'anomalia nel periodo di recessione economica, oppure nel cambiamento della tipologia di avventori (il ristorante è "scaduto" in qualità). Allo stesso modo, una diminuzione sensibile del consumo del pesce potrà essere attribuita all'aumento del suo prezzo, oppure alla divulgazione di notizie allarmistiche sull'inquinamento del mare.

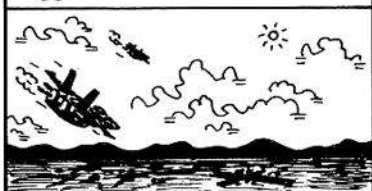
E' in questo senso, insomma, che la statistica deve essere affrontata; come aiuto, in altre parole, nella ricerca di correlazioni tra fenomeni apparentemente diversi tra loro. Elaborazioni ardite, sostenute da brillanti intuizioni, potranno comunque portare a paragonare categorie del tutto diverse tra loro, cioè non omogenee, che, tuttavia, possono presentare numerosi punti di contatto, oppure l'una esser "causa" dell'altra o viceversa.

REPERIRE I DATI

La determinazione di un semplice calcolo, e ci riferiamo alla media aritmetica, non presenta grossi problemi; si tratta, come è noto, di effettuare la somma dei vari valori in gioco e di dividerla per il loro numero. Sia che l'operazione venga condotta con carta e penna, sia che si ricorra ad un computer, la procedura si presenta lunga e noiosa ma, putroppo, non c'è altra via se non quella indicata.

Nelle applicazioni che vedremo, ovviamente(!), non saremo costretti a digitare nemmeno un dato e, addirittura, potremo approfondire il discorso sulla omogeneità dei dati di cui si occupa la statistica.

Siccome, quindi, è difficile reperire una "popolazione" (e digitarne i dati!) su cui effettuare esercizi, ci serviremo di due proce-

Novembre 1988: Primo Giovedini è riuscito finalmente a realizzare il suo sogno: partecipare alla scuola avanzata di volo e di programmazione del famoso colonnello Tom Bang. Gli fa compagnia il tenente Iceman...

```
100 REM INTRODUZIONE ALLA STATISTICA (PRG.2): INDIVIDUAZIONE
110 REM DEL NUMERO DI LOCAZIONI CONTENENTI UN VALORE
120 REM APPARTENENTE AD UN INTERVALLO SPECIFICATO
130 :
140 Y$=CHR$(147): PRINT Y$: REM CANCELLA SCHERMO
150 PRINT"QUESTO PROGRAMMA DETERMINA IL"
160 PRINT"NUMERO DI LOCAZIONI CHE CONTENGONO UN"
170 PRINT"VALORE COMPRESO TRA UN MINIMO E UN"
180 PRINT"MASSIMO IMPOSTATI DALL'UTENTE.": PRINT
190 INPUT"DALLA LOCAZIONE";LI: REM DETERMINAZIONE
200 INPUT "ALLA LOCAZIONE";LF: REM INTERVALLO.
210 TT = LF-LI+1:              REM NUMERO LOCAZIONI
220 INPUT "VALORE MINIMO";MI: IF MI<0 THEN 220
230 INPUT"VALORE MASSIMO";MA: IF MA>255 OR MA <MI THEN 230
240 PRINT Y$ "....... DEVO ARRIVARE A:" LF
250 X=0: W=0: Z=0: REM INIZIALIZZAZIONE CONTATORI
260 FOR I=LI TO LF:  REM ESAME DELLE LOCAZIONI
270 Y = PEEK(I)
280 IF (Y >= MI) AND (Y <=MA) THEN X = X + 1
290 IF Y < MI THEN W=W+1: REM VALORI MINORI
300 IF Y > MA THEN Z=Z+1: REM VALORI MAGGIORI
310 PRINT CHR$(19) I: NEXT
320 PRINT Y$ "LE LOCAZIONI ESAMINATE SONO:" TT
330 PRINT"NUMERATE DA:" LI "A:" LF: PRINT
340 PRINT X "DI QUESTE LOCAZIONI CONTENGONO UN"
350 PRINT"UN VALORE COMPRESO TRA:" MI "E:" MA: PRINT
360 PRINT W " LOC. HANNO UN VALORE MINORE DI:" MI
370 PRINT Z " LOC. HANNO UN VALORE MAGGIORE DI:" MA
380 END
```

dure. La prima prenderà in esame dati quasi casuali e ci consentirà di approfondire l'argomento sull'omogeneità; la seconda, affidata alla ricerca di numeri random, potrà farci riflettere sulla affidabilità del nostro calcolatore. Le due procedure, inoltre, verranno sviluppate, rispettivamente, in Basic e in Turbo - Pascal (versione Borland) in modo da abituare il lettore a prendere in considerazione altri linguaggi.

UNA MINIERA IN MEMORIA

Avendo a disposizione un calcolatore, spesso non si riflette abbastanza sul fatto che le sue celle di memoria rappresentano, in un certo senso, una categoria di dati ben definita. Nel caso del C/64, ad esempio, si hanno a disposizione ben 65536 locazioni (vale a dire una popolazione sufficientemente numerosa) contenenti, ciascuna, un valore intero compreso tra zero e 255.

Il primo programma proposto consente di effettuare la media aritmetica del contenuto di una zona di memoria, di cui fornire l'indirizzo di partenza (riga 170) e quello finale (180). Indicando, dopo il Run, la prima "pagina" della memoria (byte da 0 a 255) il computer giungerà ad un risultato (il nostro

Il contenuto di una particolare zona Ram può variare a causa di molteplici fattori

```
100 REM INTRODUZIONE ALLA STATISTICA (PRG. 3): INDIVIDUAZIONE
110 REM DEL NUMERO DI LOCAZIONI APPARTENENTI AD UNA
120 REM CATEGORIA MODULARE DI INTERVALLO SPECIFICATO
130 :
140 Y$=CHR$(147): PRINT Y$: REM CANCELLA SCHERMO
150 PRINT"QUESTO PROGRAMMA DETERMINA IL"
160 PRINT"NUMERO DI LOCAZIONI APPARTENENTI"
170 PRINT"AD INTERVALLI DI VALORI SUDDIVISI"
180 PRINT"SECONDO UN MODULO (FISSO) STABILITO.": PRINT
190 INPUT"DALLA LOCAZIONE";LI: REM DETERMINAZIONE
200 INPUT "ALLA LOCAZIONE";LF: REM INTERVALLO.
210 TT = LF-LI+1:              REM NUMERO LOCAZIONI
220 INPUT "INTERVALLO (1/128)";IN: IF IN<1 OR IN >128 THEN 220
230 X1=INT(255/IN): DIM W(X1)
240 PRINT Y$ "....... DEVO ARRIVARE A:" LF
250 FOR I=LI TO LF:  REM ESAME DELLE LOCAZIONI
260 Y = PEEK(I): X=Y/IN
270 W(X) = W(X) + 1: REM INCREMENTO
280 PRINT CHR$(19) I: NEXT
290 PRINT Y$ "LE LOCAZIONI ESAMINATE SONO:" TT
300 PRINT"NUMERATE DA:" LI "A:" LF: PRINT
310 FOR Y=0 TO X1: PRINT "GRUPPO:" TAB(7)Y*IN;
320 IF (Y+1)*IN-1<256 THEN PRINT TAB(12) (Y+1)*IN-1;
330 PRINT TAB(17)" CONTENUTO:" W(Y): NEXT
340 END
```

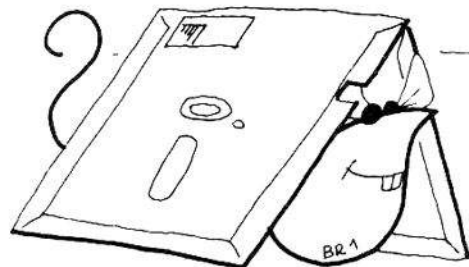
Un numero, per esser considerato casuale, deve possedere caratteristiche ben precise

C/64 ha visualizzato 60.4296, il vostro potrà fornire un valore diverso) che può significar qualcosa ma... è destinato, prevalentemente, a non significare proprio nulla.

Tanto per iniziare, una locazione può contenere solo valori interi: il valore decimale, che invece risulta, non si inserisce correttamente nella "logica" del computer. Inoltre, il valor medio può cambiare, e di molto, a seconda della zona di memoria presa in esame.

Anche se il lettore non ne conosce il contenuto, la cosiddetta "mappa della memoria" del C/64 contiene, a seconda delle zone, valori ben precisi ed immutabili (le Rom) come pure valori indefiniti (zona Ram del Basic, alterata automaticamente da varie elaborazioni) oppure settabile a volontà (Ram video, schermo Hi-Res) dall'utente il quale può, magari, inserire routine l.m. oppure valori definiti, o casuali, nella Ram a disposizione.

L'invito ad "esplorare" la memoria del computer, e ad effettuare la media dei valori contenuti, ha il solo compito di far riflettere sul concetto di "popolazione" e sulla omogeneità dei dati. Se ad esempio, e concludiamo con il primo listato, si determina la media tra le locazioni 1024 e 2023 (cioè l'area video) si noterà che il risultato vale 31.936 dal momento che, durante l'elaborazione, la maggior parte dell'area video risulta cancellata (codice schermo = 32) e compare solo il messaggio...
"XXX.. devo arrivare a: 2023"
...i cui caratteri, appunto, alterano la media

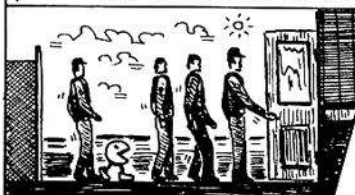
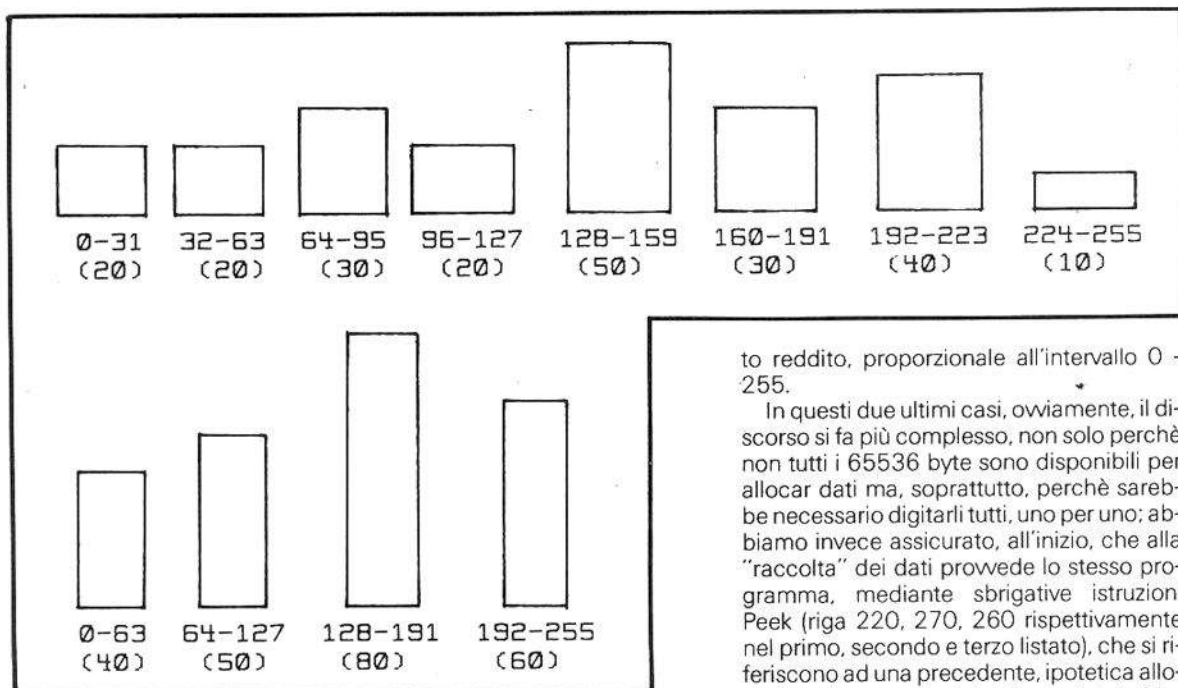
```
PROGRAM primi_passi_con_la_statistica;
 VAR              x1: real;
       lf, x, y, li , iv: integer;
 VAR mat: array [0..255] of integer;
PROCEDURE azzeramento; (* Azzera ciascun elemento del vettore *)
  begin; li: =0; repeat; mat[li]: =0; li: =li+1; until li=256;
  end;
PROCEDURE vettore; (* Genera un numero e lo assegna all'elemento del vettore*)
   begin; li: =0; repeat; y: =random (256); x: =y div iv; mat[x]: =mat[x]+1; li: =li+1;
                  until li=lf;
   end;
 BEGIN clrscr; azzeramento;  (* Esegue prima procedura = subroutine *)
 writeln('Questo programma genera X numeri casuali (ciascuno compreso tra',
        ' zero e 255 e in quantita'' non maggiore di 32767) e li raggruppa',
        ' in "classi" di numeri che, alla fine, vengono visualizzate');
 writeln;
 repeat;
  write ('Quanti numeri casuali? '); readln(lf);
 until lf>0;
 repeat;   write ('Intervallo desiderato (0-255)? '); readln(iv);
 until (iv>0) and (iv<256);
 writeln; x1: =255/iv;
 vettore; (* Esegue seconda procedura *) y: =0;
 repeat;
   write('Intervallo: ',  y*iv:3); (* Formatta output a 3 cifre *)
     if (y+1)*iv-1  <256
     then write(' - ',(y+1)*iv-1:3)
     else write(' - 255'); (* "Messaggio" alternativo se limite max.  255 *)
  writeln(', contenuto: ',mat[y]); y: =y+1; write(^g);
 until y > x1;                                          (* ^g    segnale sonoro *)
 END.
```

che, in caso di schermo totalmente cancellato, sarebbe di 32 (provate infatti con l'intervallo 1064 - 2023). L'omogeneità dei dati, legata a varie congetture sulla zona della popolazione presa in esame, porta a considerazioni diverse, insomma, a seconda dei casi che possono presentarsi.

IL SECONDO LISTATO

Abbiamo quindi "scoperto" (ma lo sapevamo di già) che può non aver senso determinare la media di una zona di memoria. Ci accorgiamo, però, che potrebbe essere interessante individuare il numero di locazioni che contengono un valore ben preciso, oppure appartenente ad un intervallo delimitato.

Supponiamo di aver usato un word processor, e che questo abbia memorizzato (secondo il codice Ascii) il testo nella Ram numerata da 49152 a 50000. Volendo sapere quante volte sono stati battuti i caratteri "A", "B" e "C" non dovremo far altro (dopo il Run) che indicare l'inizio (49152) e la fine (50000) dell'area che interessa; rispondere con 65 (codice Ascii di "A"), alla richiesta sul valore minimo, e 67 (Ascii di "C") a quella sul valore massimo. Dopo un arco di tempo proporzionale all'area esplorata comparirà la "statistica" desiderata, sotto forma di numero di celle contenenti uno dei tre valori e, con semplici sottrazioni, quelle con valori più alti di 67 e più bassi di 65.

Volendo ricercare un solo valore (ad e-

Dopo la formulazione di un problema, è necessaria la sua "traduzione" nel linguaggio di programmazione desiderato

Il Pascal è un linguaggio che ben si presta per risolvere problemi che richiedono procedure ricorsive.

sempio: quante locazioni contengono il valore 125?) sarà sufficiente digitare 125 sia alla prima, sia alla seconda domanda.

Le risposte ottenibili, nel tipo di ricerca che il secondo listato può condurre, possono essere utili in molti casi. Facendo finta, ad esempio, che i dati presi in esame non sono relativi alla memoria di un computer, ma ai punti di rilevamento ecologici di una regione geografica, si potrebbe "estrarre" il numero di campioni di aria inquinata il cui valore può oscillare (guarda caso!) tra 0 e 255.

Oppure (sempre considerando valida la proporzionalità minimo - massimo con i due estremi 0 - 255) considerando una città di 65536 abitanti che producono un certo reddito, proporzionale all'intervallo 0 - 255.

In questi due ultimi casi, ovviamente, il discorso si fa più complesso, non solo perchè non tutti i 65536 byte sono disponibili per allocar dati ma, soprattutto, perchè sarebbe necessario digitarli tutti, uno per uno; abbiamo invece assicurato, all'inizio, che alla "raccolta" dei dati provvede lo stesso programma, mediante sbrigative istruzioni Peek (riga 220, 270, 260 rispettivamente nel primo, secondo e terzo listato), che si riferiscono ad una precedente, ipotetica allocazione in memoria dei dati su cui si desidera operare.

E' ovvio che i lettori più volenterosi potranno generare un vettore numerico (con il comando DIM) ed inserire una routine di raccolta manuale di dati (For... Input... Next) da esplorare successivamente.

Per ciò che ci riguarda, invece, il nostro desiderio era limitato all'indicazione di alcune vie da seguire per fare i primi approcci con la statistica.

IL TERZO LISTATO

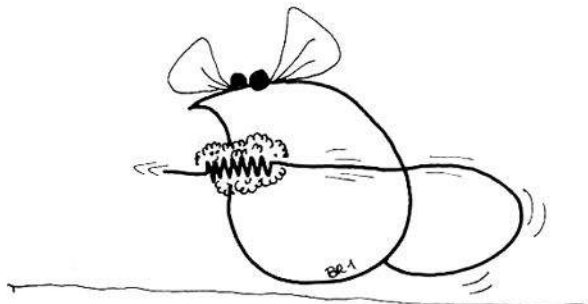
I due istogrammi riportati nelle figure di queste pagine sono relativi al raggruppamento degli stessi dati che supponiamo di

aver rilevato in precedenza. Nella prima figura notiamo la suddivisione, degli stessi dati, in otto gruppi (0 - 31, 32 - 63 fino a 224 - 255). Al primo appartengono 20 dati, come anche al secondo, fino all'ultimo che rappresenta 10 dati con valore compreso tra 224 e 255.

Pur se, da un punto di vista strettamente logico, gli istogrammi della prima figura sono "veri", tuttavia non riusciamo ad individuare informazioni utili su eventuali "tendenze": i dati stessi, raggruppati in un numero eccessivo di intervalli, appaiono come un susseguirsi di rettangolini praticamente eguali tra loro.

Se, invece, raggruppiamo gli stessi dati in soli 4 gruppi, come indicato in figura 2, è più agevole assimilare la distribuzione degli stessi ad un andamento a campana, parente prossimo di una più rigorosa distribuzione gaussiana. Nella figura 2, insomma, è più agevole individuare similitudini (e differenze) dei dati a disposizione, a patto, ovviamente, di aver sufficiente esperienza (o intuito) per farlo.

Il terzo listato esegue la suddivisione dei dati (che sono le solite locazioni di memoria del nostro Commodore), in classi definibili dall'utente.

Dopo il Run sarà necessario indicare la prima e l'ultima locazione su cui effettuare la ricerca; subito dopo verrà richiesto il "modulo", cioè l'intervallo dei valori in cui suddividere la stessa ricerca. Provate a suddividere, una stessa area, dapprima in due intervalli, (rispondendo con 128 alla terza domanda), poi con 64, poi con 32 e così via fino ad individuare un output contenente informazioni che possono essere considerate di un certo interesse. Se l'intervallo non è modulare con una potenza di 2, l'ultimo "raggruppamento", può presentare anomalie dal momento che si riferisce ad un gruppo di valori minore dei precedenti.

Vi potete divertire, ad esempio, a verificare la presenza, nelle Rom, di codici macchina ricorrenti effettuando varie statistiche che sarete in grado di individuare. Anche in questo caso potrete immaginare di avere a che fare con dati relativi a popolazioni di tipo ben diverso dai chip di un elaboratore.

IL CASO NEL FORMATO TURBO - PASCAL

Il contenuto della memoria di un computer, come già detto, non può certo esser considerato casuale, a dispetto del numero relativamente elevato di dati a disposizione.

I numeri, infatti, possono esser considerati casuali solo se, presi in gran quantità, e suddivisi in qualsivoglia intervallo, generano istogrammi di altezza praticamente identica.

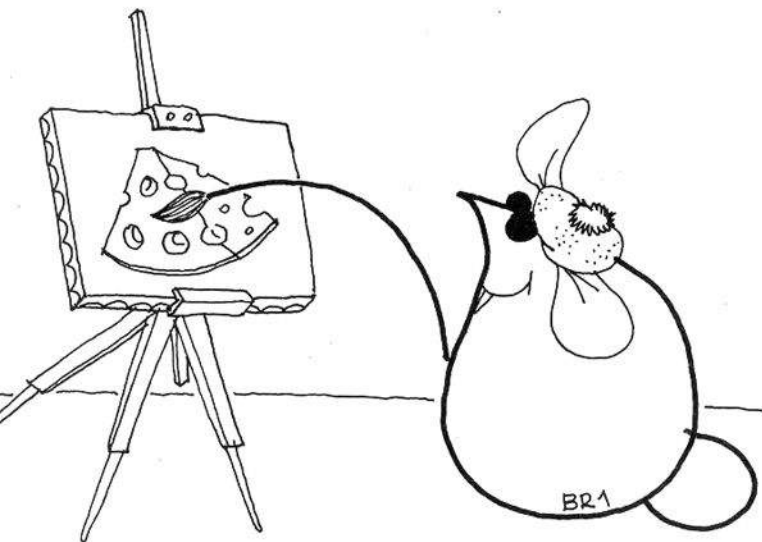
Quasi tutti i linguaggi offrono la possibilità di generare numeri casuali. Il problema è quello di verificare se, in effetti, offrono la caratteristica della completa casualità. Un metodo per stabilire la bontà della procedura tipica di un certo linguaggio è certamente quello di far generare alcune decine di migliaia di valori casuali (appartenenti ad un intervallo prefissato) da suddividere, in varie fasi successive, in più classi omogenee. I valori che risulteranno alla fine dell'elaborazione dovranno essere sempre identici o, quantomeno, risultare di poco diversi tra loro.

DENTRO I PROGRAMMI

Sui programmi in Basic non c'è molto da dire; si fa ricorso al solito ciclo For...Next per la lettura dei dati e vengono utilizzate variabili (X, Y, W e simili) che hanno il compito di conteggiare i valori, via via individuati, appartenenti alle varie classi.

Banale è la formattazione dell'output che ricorre al TAB per un (approssimativo) incolonnamento dei risultati. Facili da interpretare, infine, i messaggi che compaiono sul video.

Il programma in Turbo Pascal utilizza prevalentemente variabili di tipo intero e sfrutta la capacità di generare un numero casuale maggiore di zero e minore di 256. La matrice di nome "Mat" non viene sempre sfruttata per intero ma ciò che conta è il risultato. L'indentazione impostata dovrebbe facilitare la lettura del listato, in cui compaiono due procedure, forse non indispensabili.

Ai lettori, ovviamente, il divertente compito di apportare modifiche di vario tipo, in accordo alle proprie esigenze.

Il programma in Turbo Pascal Borland riportato in queste pagine (che gira allo stesso modo sia nella versione Ms-Dos sia sul C/128 in modalità CP/M) richiede il numero di valori casuali da generare (non maggiore di 32737) e, subito dopo, l'intervallo in cui suddividere gli stessi valori, dopo averli determinati. Anche in questo caso è stato mantenuto lo stesso intervallo già esaminato con i programmi in Basic (0 - 255) per confrontare, però, due situazioni totalmente diverse: nei programmi in Basic si esaminava il contenuto di varie locazioni che non era possibile definire casuale; stavolta, invece, dovremmo avere a che fare con numeri casuali per... definizione. Il lettore potrà verificare che, con elaborazioni riferite ad un numero sufficientemente elevato di dati, i valori generati appartengono, in effetti, a classi omogenee tra loro. Operando con altre versioni di Pascal (o con vari elaboratori?) possono verificarsi discontinuità più o meno evidenti.

Inutile sottolineare che, impostando intervalli non modulari (rispondendo, ad esempio, con 77 alla relativa domanda) si verificano anomalie dovute, però, alla scorretta impostazione del procedimento.

VIAGGIO AL CENTRO DEL BASIC

Alcune Sys piuttosto utili per chi è abituato a lavorare sia in Basic sia in linguaggio macchina con il piccolo computer Commodore

di **Antonio Pastorelli**

Anche il C/16 (ed il Plus/4) consente di operare in l.m. con la massima facilità

Il microprocessore di un qualsiasi computer è in grado di interpretare determinati codici binari, che corrispondono a istruzioni Assembly.

Di conseguenza l'unico "vero" linguaggio è l'Assembly (o, impropriamente, Assembler); ogni altro linguaggio è interpretato oppure compilato.

Per poter operare in ambiente Basic, esistono, su ROM, routine in linguaggio macchina (l.m.) che permettono di editare il programma e di riconoscere ed eseguire le varie istruzioni.

In questo articolo daremo uno sguardo ad alcune di esse che eseguono comandi Basic, fornendo l'indirizzo di partenza e la sintassi da usare per utilizzarle correttamente.

Innanzitutto facciamo conoscenza con gli autori del Firmware del nostro C/16 e Plus/4, impartendo il seguente comando, in modo diretto...

SYS 52650

...che "lancia" la routine in l.m. pubblicata a parte.

Come si può notare dal disassemblato commentato, gli autori, pur avendo inserito i loro nomi in memoria, hanno cercato di nasconderli, codificando i valori Ascii delle singole lettere, invertendo i bit 0, 2, 4 e 6, con EOR #$55.

Il successivo EOR #$55, della routine presentata, ripristina il corretto valore nei bit citati del codice Ascii di ciascuna lettera, e quindi la stampa su video.

Il Basic 3.5, contrariamente alla versione 2.0 implementata sul C/64, è fornito del comando Directory, che permette di visualizzare il catalogo dei file presenti su disco, senza cancellare il programma in memoria.

Tale routine parte dalla locazione 51388 e può quindi essere chiamata da Basic con:

SYS 51388 (Lista tutta la directory)
SYS 51388 "S*" (Lista l'elenco dei file il cui nome inizia per "S")

Pertanto da l.m. è possibile utilizzare la routine di Directory mediante l'istruzione:

JSR $C8C3

Analogamente, se state lavorando con il Monitor incorporato, con il comando "G C8C3", otterrete egualmente la directory.

Ecco, di seguito, gli indirizzi di altre routine relative a noti comandi Basic:

COLOR	50458	GET	37048	DLOAD	51537
LET	36476	READ	37199	DSAVE	51521
PRINT	36861	RESTORE	35994	FOR	44490
PRINT#	36832	POKE	40466	NEXT	37526
INPUT	37128	WAIT	40554	LOOP	46642
INPUT#	37102	REM	36363	HELP	46824
ON	36379	DEF	39581	SCALE	50616
GOTO	36170	END	36058	LOCATE	50447
GOSUB	36140	SCNCLR	50535	DRAW	50393
RETURN	36227	RENUMBER	3919	SSHAPE	48681
LOAD	42995	DELETE	44634	GSHAPE	48437
SAVE	42974	MONITOR	65362	CHAR	47842
VERIFY	42992	BACKUP	51712	PAINT	47313
DO	46423	RENAME	51700	AUTO	46797
IF	36321	COPY	51673	VOL	47293
GRAPHIC	50627	COLLECT	51660	SOUND	47177
CIRCLE	49182	SCRATCH	51612	TROFF	46677
BOX	47313	HEADER	51560	TRON	46674
				TRAP	46123

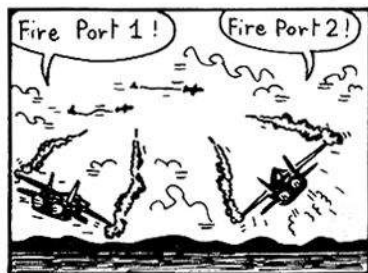
E' sufficiente impartire la Sys "giusta", corredata da eventuali parametri, per sortire gli effetti più disparati

E' importante sottolineare che le istruzioni SYS, relative a comandi che richiedono un primo paramentro numerico, vanno digitate facendo uso delle parentesi, come ad esempio:

SYS (47313) 1, 10, 5, 40, 75

In caso contrario il primo parametro numerico potrebbe confondersi con l'indirizzo indicato dalla SYS.

La conoscenza degli Entry-point delle routine del Basic, è un utile strumento per il raggiungimento di scopi diversi.

E' possibile utilizzare alcune routine del Basic direttamente da l.m, evitando di doverle riscrivere, risparmiando così sia su tempo e difficoltà di programmazione, sia sulla lunghezza del programma, dal momento che tali routine sono già residenti su ROM.

Chi scrive programmi in Basic e desidera occultare particolari procedure, algoritmi e così via, può far uso di istruzioni SYS in altenativa alla sintassi standard; chi vorrà decifrarle dovrà sudare le famose sette camicie.

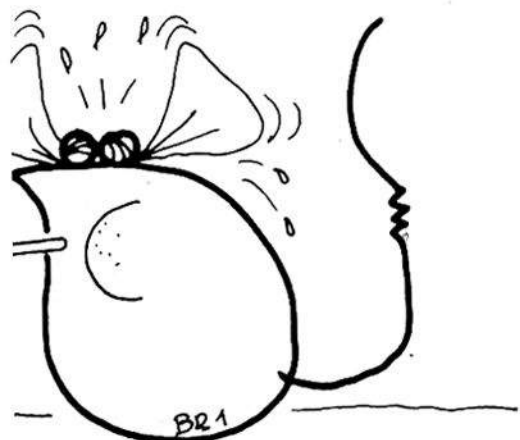
L'ultima applicazione, ma non per questo la meno importante, è quella didattica.

Conoscendo l'entry-point di una routine del Basic, e la sintassi del relativo comando, sarete facilitati nella comprensione di un eventuale disassemblato; potrete quindi studiare come il Basic controlla la sintassi dei comandi e come li esegue; poi, presa una certa confidenza con il l.m, riuscirete ad implementare nuovi comandi Basic.

Il programma di queste pagine mostra il corretto utilizzo di alcune routine, disegnando il simbolo delle olimpiadi all'interno di una bandiera colorata.

```
CDAA  CLD              ;Elimina modo
                       ;decimale
CDAB  LDY #$21         ;Y=33 ($21)
CDAD  LDA $CD89,Y      ;carica codici da
                       ;$CD89 in poi
CDB0  EOR #$55         ;decodifica con $55
CDB2  JSR $FFD2        ;stampa lettera
                       ;decodificata
CDB5  DEY              ;Y=Y-1
CDB6  BPL $CDAD        ;Salta a CDAD se
                       ;Y=>0
CDB8  RTS              ;RITORNA
---------------------------------------------
```

Ecco la routine, contenuta nel C/16, che visualizza i nomi degli autori del firmware.

```
1 SYS 36363 DEMO SYS-BASIC PER C/16
2 SYS 36363 BY ANTONIO PASTORELLI
3 :
10 REM FISSA COLORI INIZIALI
20 REM CON IL COMANDO COLOR
30 :
40 SYS (50458) 0,1:SYS (50458) 4,1
45 SYS (50458) 1,2:REM COLOR
50 :
60 SYS (50627) 1,1:REM GRAPHIC
70 SYS (50458) 1,3,5:REM COLOR
80 SYS (49182) 1,50,50,10,10:REM CIRCLE
90 SYS (49182) 1,65,50,10,10:REM CIRCLE
100 SYS (49182) 1,80,50,10,10:REM CIRCLE
110 :
120 SYS (49182) 1,58,65,10,10:REM CIRCLE
130 SYS (49182) 1,73,65,10,10:REM CIRCLE
140 :
150 SYS (47842) 1,20,30,110,85:REM BOX
160 SYS (47313) 1,21,31:REM PAINT
170 :
180 SYS (40466) 239,0:REM POKE 239,0
190 SYS (40554) 239,1:REM WAIT 239,1
200 SYS (50627) 0:REM GRAPHIC
210 SYS 50535: REM SCNCLR
```

UN ANTILIST A PROVA DI HACKER

Aggiungiamo, ai nostri programmi Basic, un pizzico di cattiveria; naturalmente, in Assembly

di **Domenico Pavone**

PROTEGGERE IN ASSEMBLY

Le tecniche di protezione di un programma, tema ricorrente nelle pagine di questa rivista, risultano certamente uno degli argomenti più sentiti per l'eletto popolo dei commodoriani, vorace divoratore di magiche cartucce, sofisticati Error maker, e chi più ne ha più ne metta.

L'uso di mezzi così evoluti, d'altra parte, più che comprensibile nel caso di megaprogrammi costati mesi di fatica (altrui), spesso risulta sproporzionato per applicazioni più modeste, ma non per questo meno importanti. Può essere il caso di un programmino in Basic, magari lungo solo qualche decina di righe, ma pur sempre frutto delle... illuminate intuizioni del nostro genio (quando si dice la modestia!).

In casi simili, risulta spesso più pratico limitarsi a scoraggiare eventuali curiosi con sistemi di protezione più semplici e meno ingombranti, ben consapevoli, in ogni caso, della "Prima Legge dell'Hacker", che più o meno suona così: "Tutto ciò che è stato protetto, può (deve?) essere sprotetto".

Purchè, c'e da aggiungere, si disponga delle appropriate conoscenze. In altre parole, purchè si sappia usare a dovere il linguaggio macchina.

Ed ecco che giungiamo all'obiettivo di questo articolo: fornirsi di un antilist semplice ma un po' diverso dal solito, il più possibile personalizzato con l'aiuto del Macro Assembler Commodore e, soprattutto, che ci fornisca l'occasione per approfondire alcuni aspetti della programmazione in Assembly.

Molti di voi già sapranno che, per rendere inattivo il comando List, è sufficiente modificare il contenuto delle locazioni di memoria 774 e 775, inserendo le Poke necessarie nel programma che si intende proteggere da occhi indiscreti (magari dotandolo anche di autostart).

Basta già impartire...

Poke 774,116: Poke 775,164

A sera, viene resa nota la classifica provvisoria del Trofeo Tom Bang: in testa sono Giovedini ed Iceman, a pari merito, poi seguono Hang On, Anthony Past O'Really, Michael Mays, Pacman e tutti gli altri...

PRINCIPALI INDIRIZZAMENTI DEL 6510

IMPLICITO	TAX	Non necessita di un operando.
IMMEDIATO	LDA #$FF	L'operando rappresenta un numero (255) che viene caricato in accumulatore.
ASSOLUTO	LDX $8000	L'operando indica la locazione di memoria 32768, il cui contenuto viene caricato in X.
ASSOLUTO INDICIZZATO	STA $8000,X	Il contenuto di X viene sommato a 32768 per determinare la locazione in cui depositare il contenuto dell'accumulatore.
INDIRETTO	JMP ($306)	Salta all'indirizzo specificato dalle locazioni $306 e $307 (nel formato basso/alto).
INDIRETTO POSTINDICIZZATO CON Y	LDA ($FB),Y	Carica in accumulatore il contenuto della locazione puntata da $FB ed $FC, al cui indirizzo deve essere sommato Y. Obbligatorio che il puntatore (nel nostro caso $FB) si trovi in pagina zero.
INDIRETTO PREINDICIZZATO CON X	LDA ($FB,X)	Il valore di X viene sommato ad $FB (non al suo contenuto!) per indicare il primo dei due puntatori (in pagina zero) contenenti l'indirizzo della locazione col cui contenuto caricare A.
RELATIVO	BEQ LABEL	Salto condizionato all'indirizzo indicato da LABEL, che deve essere compreso tra -127 e +128 byte rispetto all'indirizzo corrente.

...per inibire la visualizzazione di un listato. Se, però, vogliamo realizzare qualcosa che renda un po' più difficile la vita di eventuali spioni, è importante capire cosa avvenga realmente allorchè si manipolano le due suddette locazioni.

A tale scopo, niente di meglio che farci dare una mano dal Macro Assembler, sfruttando i suoi monitor per una breve escursione all'interno del nostro C/64. Le indicazioni che seguiranno, comunque, risulteranno valide anche utilizzando altri programmi di Monitor. Inserite dunque il dischetto Macro Assembler nel drive, quindi digitate:

Load "MONITOR$C000",8,1

A caricamento ultimato, impartite SYS 49152.

Prima di proseguire, occorre precisare che la routine di sistema che presiede all'esecuzione del comando List è allocata, nella Rom riservata all'interprete del Basic, a partire dalla locazione di memoria 42652, corrispondente, in esadecimale, ad $A69C. Poichè la notazione esadecimale è l'unica utilizzabile in ambiente monitor, digitate...

D A69C A719

...che sta per Disassembla (D) dalla locazione numero 42652 (esad.=$A69C) alla locazione 42777 (esad.=$A719).

Applicare una tecnica di protezione è un'ottima scusa per approfondire l'uso dell'Assembly

L'indirizzamento indicizzato è un efficiente metodo per caricare, o trascrivere, i contenuti di varie locazioni

Dopo il return, vedrete scorrere velocemente sullo schermo una sequenza di istruzioni Assembly che, senza scendere nei particolari, servono al sistema per settare alcuni parametri necessari alla routine di List.

Fermiamo però l'attenzione sull'ultima riga del disassemblato, che, se si è proceduto correttamente, dovrebbe leggersi così:

A7176C 06 03JMP ($0306)

Abbiamo già visto, nei precedenti articoli sull'Assembly, come l'istruzione JMP (JuMP = salta) equivalga al Basic GOTO, con l'operando che indica l'indirizzo al quale saltare. Se, però, l'operando è racchiuso in parentesi, come nel nostro caso, ci troviamo di fronte ad un diverso tipo di indirizzamento, chiamato "Indiretto", che andrà interpretato come segue: salta alla locazione il cui indirizzo è memorizzato in due celle contigue di memoria (chiamate "puntatori"), la prima delle quali è indicata dall'operando stesso.

La routine di List, quindi, con l'istruzione appena vista, proseguirà all'indirizzo contenuto (nalla solita forma byte basso/alto) nelle locazioni $306 e $307, in decimale 774 e 775. Andiamo quindi a vedere quale sarà questo benedetto indirizzo, sfruttando ancora il monitor del Macro Assembler.

Premete due volte il tasto Return, quindi digitate M 0306 (+ return) e vedrete apparire (M = Memory) sullo schermo i valori che interessano, ovvero i primi due della riga apparsa: 1A e A7. Poichè gli indirizzi sono normalmente memorizzati nel formato basso/alto, il programma l.m. giunto all'istruzione $A717 (vedi prima) salterà alla locazione $A71A; l'istruzione da eseguire immediatamente successiva all'istruzione di salto contenuta in $A717 sarà, insomma, quella contenuta nell'indirizzo A71A. Questa (routine di List) stampa sullo schermo il listato del programma Basic presente in memoria.

E' questo un tipo di procedura riscontrabile anche in molte altre routine del sistema operativo, progettata proprio per consentire al programmatore di apportarvi eventuali modifiche.

VIETATO LISTARE!

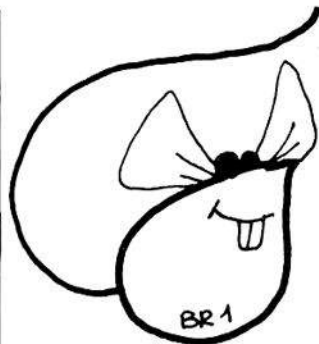
Se, come è noto, non è possibile intervenire sulle locazioni poste in Rom, possiamo tuttavia agire sui puntatori (nel nostro caso le locazioni 774 e 775), situati in memoria di tipo Ram, per dirottare la prosecuzione della routine dove più ci aggrada.

Le semplici Poke prima accennate, per esempio, non faranno altro che far continuare il normale List all'indirizzo decimale 116 + 164 * 256 = 42100, cui corrisponde una sequenza l.m. posta in Rom che semplicemente provoca un ritorno al Basic stampando il prompt READY.

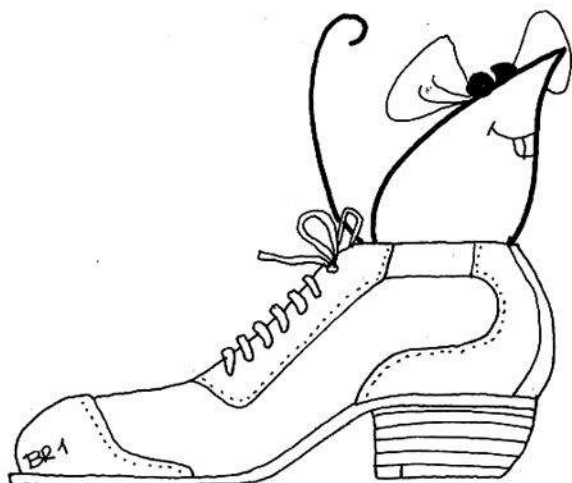
Nulla vieta, come forse avrete già intuito, l'inserimento nei due puntatori dell'indirizzo di una nostra routine, che ovviamente dovrà operare esclusivamente in linguaggio macchina.

Vediamolo in pratica, cominciando col resettare il computer, quindi copiando attentamente e salvando il breve listato proposto in queste pagine. Una volta mandato in esecuzione, questo insedierà in memoria un programma l.m. a partire dalla locazione 679, cioè una piccola zona di memoria libera da interferenze che si estende fino alla locazione 767.

La scelta di quest'area non è casuale, in quanto presenta una caratteristica: viene "ripulita" (ponendo a zero le sue celle) dal sistema operativo durante la procedura di

```
01 0000                  ;-------------------------------------------
02 0000                  ;    DISASSEMBLATO IN FORMATO MACROASSEMBLER
03 0000                  ;-------------------------------------------
04 0000                  BASIC  =$2B           ;dec.= 43
05 0000                  PUNT   =$FB           ;dec.= 251
06 0000                  CHROUT =$FFD2         ;dec.= 65490
07 0000                  READY  =$A474         ;dec.= 42100
08 0000                  RESET  =$FCE2         ;dec.= 64738
09 0000                  LIST   =$306          ;dec.= 774
10 0000                         *=$2A7         ;dec.= 679
11 02A7                  ;-------------------------------------------
12 02A7  A9 02                  LDA #>START   ; Pone l'indirizzo $2B6
13 02A9  A2 B6                  LDX #<START   ; nel vettore della routine
14 02AB  8E 06 03               STX LIST      ; di sistema LIST.
15 02AE  8D 07 03               STA LIST+1    ;
16 02B1  A2 FE                  LDX #$FE      ; Carica la locazione 2
17 02B3  86 02                  STX $02       ; con il valore 254.
18 02B5  60                     RTS           ; Ritorna al basic.
19 02B6                  ;-------------------------------------------
20 02B6  E6 02           START  INC $02       ; Incrementa val. locaz. 2.
21 02B8  F0 0F                  BEQ CANCEL    ; Se loc.2 = 0, salta.
22 02BA  A2 00                  LDX #00       ; Indice X = 0.
23 02BC  BD E5 02        AVVISO LDA MESSG,X   ; Legge carattere.
24 02BF  C9 00                  CMP #00       ; Controlla se zero.
25 02C1  F0 1F                  BEQ USCITA    ; Se = 0, salta.
26 02C3  20 D2 FF               JSR CHROUT    ; Stampa carattere.
27 02C6  E8                     INX           ; Incrementa indice X.
28 02C7  D0 F3                  BNE AVVISO    ; Torna a leggere.
29 02C9                  ;-------------------------------------------
30 02C9  A9 00           CANCEL LDA #00       ; Prepara due locazioni di
31 02CB  85 FB                  STA PUNT      ; pagina zero con indirizzo
32 02CD  A6 2E                  LDX BASIC+3   ; di fine del programma
33 02CF  86 FC                  STX PUNT+1    ; basic corrente.
34 02D1  A0 FF                  LDY #$FF      ; Indice Y = 255.
35 02D3  91 FB           LOOP   STA (PUNT),Y  ; Azzera loc. puntata + Y.
36 02D5  88                     DEY           ; Y = Y - 1.
37 02D6  D0 FB                  BNE LOOP      ; Se Y <> 0, salta.
38 02D8  C6 FC                  DEC PUNT+1    ; Decrementa puntatore alto.
39 02DA  CA                     DEX           ; X = X - 1.
40 02DB  E0 07                  CPX #07       ; Compara X con 7.
41 02DD  D0 F4                  BNE LOOP      ; Se X <> 7, salta.
42 02DF  4C E2 FC               JMP RESET     ; Reset del sistema.
43 02E2                  ;-------------------------------------------
44 02E2  4C 74 A4        USCITA JMP READY     ; Ritorno al basic.
45 02E5                  ;-------------------------------------------
46 02E5  4E 4F           MESSG  .BYTE 'NON RIPROVARCI!',0
47 02F4  00
```

reset. Vedremo meglio tra breve come ciò può tornarci utile.

Impartiamo dunque il RUN, dopodichè attiviamo la routine con SYS 679. Se ora si proverà a chiedere il List, per tutta risposta avremo un sibillino avvertimento: NON RIPROVARCI!

Non dando ascolto al gentile consiglio, riproviamo con List (e return): tutto ciò che otterremo sarà un reset completo del siste-

```
100 REM ----------------------------
110 PRINT"ANTILIST (QUASI) CATTIVO"
120 REM ----------------------------
130 PRINT"   ATTIVARE CON:  SYS 679"
140 REM ----------------------------
150 :
160 FOR X=0 TO 77:READ A: B=B+A
170 POKE 679+X,A: NEXT
180 IF B=10057 THEN END
190 PRINT "ERRORE NEI DATA!":END
200 :
210 DATA 169,002,162,182,142,006,003
220 DATA 141,007,003,162,254,134,002
230 DATA 096,230,002,240,015,162,000
240 DATA 189,229,002,201,000,240,031
250 DATA 032,210,255,232,208,243,169
260 DATA 000,133,251,166,046,134,252
270 DATA 160,255,145,251,136,208,251
280 DATA 198,252,202,224,007,208,244
290 DATA 076,226,252,076,116,164,078
300 DATA 079,078,032,082,073,080,082
310 DATA 079,086,065,082,067,073,033
320 DATA 000
330 PRINT"PROVA A DIGITARE: LIST"
340 END
```

Il salto indicizzato consente di manipolare indirizzi residenti parte in Rom e parte in Ram

ma, almeno apparentemente. In effetti, è avvenuto qualcosa di più. Per verificarlo ricorriamo ancora al monitor, procedendo come segue:

1) Caricate nuovamente in memoria il "MONITOR$C000" ed attivatelo.

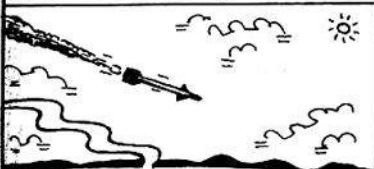
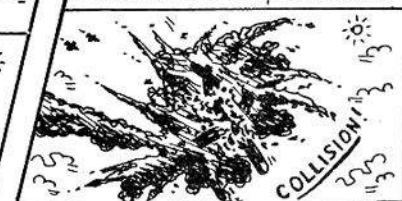
2) Impartite il comando I 0801, che mostrerà le prime otto locazioni dell'area Basic: conterranno solo degli zeri; come potrete constatare utilizzando il tasto cursore verso il basso, anche le locazioni successive risulteranno azzerate.

3) Date ora il comando D 02A7 02F4, che dovrebbe mostrare il disassemblato della nostra routine antilist ($2A7 = decimale 679): anche qui non vedremo altro che una bella sfilza di zeri.

Cos'è avvenuto? Semplice: Il reset ha azzerato le locazioni dove era presente la nostra routine, la quale però, prima di... defungere, ha provveduto a riempire di zeri l'area precedentemente occupata dal programma Basic in memoria, lasciando con un palmo di naso i curiosi più smanettoni.

Un semplice reset, infatti, non cancella le locazioni della Ram Basic, che risulta quindi ugualmente leggibile facendo uso di un monitor. Provate a verificarlo: se avete seguito con attenzione quanto finora esposto, le operazioni necessarie non dovrebbero ormai rappresentare un problema.

LA ROUTINE

E se qualcuno, ignorando quel residuo barlume di bontà (sigh!) che ci ha spinti ad

inserire l'avvertimento, intervenisse sui puntatori prima di effettuare un secondo list?

Di fronte ad un dubbio simile, la risposta non potrà che essere una: rendere ancora più cattiva la nostra routine eliminandone la prima parte, o aggiungervi ulteriori elementi di nostra ideazione (= guerra ad oltranza).

Qualunque intervento, però, passa necessariamente attraverso un'elaborazione in Assembly, per cui vediamo di esaminare più attentamente l'antilist appena sperimentato riferendoci al disassemblato di queste pagine, realizzato con il Macro Assembler Commodore. Starà poi a voi, magari sfruttando la semplicità d'uso dello stesso, adattare la protezione alle vostre (maligne) esigenze.

Il primo gruppo di istruzioni (righe 12 - 18) si limita a depositare, nel vettore di List, (le solite locazioni 774 e 775) l'indirizzo della nostra routine, nella fattispecie il valore esadecimale $02B6, individuato dall'etichetta START di riga 20. Come si può notare, nelle linee 12 e 13 viene usato un modo particolare per specificare l'operando, estremamente utile per rendere totalmente trasportabile la routine stessa.

L'indirizzo di cui sopra, infatti, va considerato nei due byte che lo compongono: quello basso, B6, e quello alto, 02. Potremmo usare direttamente questi valori per caricare i registri A ed X di riga 12 e 13, ma, se volessimo poi trasferire altrove la routine, dovremmo adattare i due operandi alla nuova allocazione.

Per fortuna il Macro Assembler ci viene in soccorso. Adoperando infatti il simbolo "maggiore", posizionandolo prima di un'etichetta (come in riga 12), ne verrà considerato il byte alto, mentre il simbolo "minore" farà riferimento al suo byte basso. In pratica, nel nostro antilist, il primo operando equivarrà a 02, mentre il secondo verrà interpretato dall'assembler come il numero esadecimale B6.

Con questa facilitazione, per modificare l'allocazione della routine, basterà cambiare solamente il valore posto dopo il simbolo "*=" (vedi riga 10), tutto il resto si adeguerà di conseguenza. Più comodo di così...

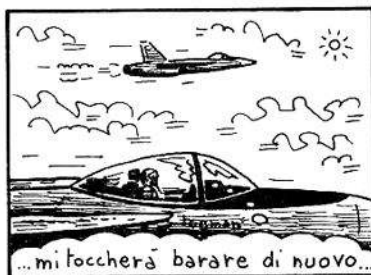
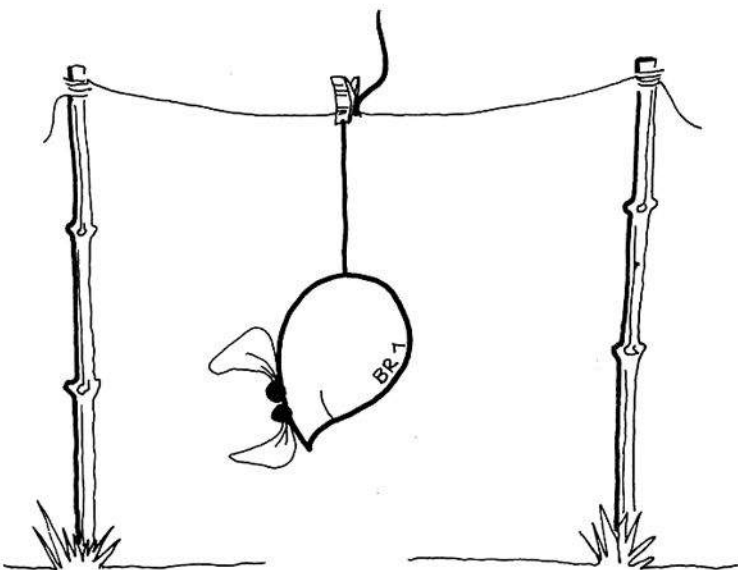
Ma torniamo a noi. In riga 16 viene caricata la locazione 2 con il valore $FE (decimale 254), in modo che funga da contatore. Quando verrà impartito il comando List, infatti, per prima cosa si incrementerà il contenuto di questa locazione (riga 20), che assumerà dunque il valore 255. Un ulteriore incremento (= la seconda volta che si dà il List), azzererà la locazione.

Il BEQ (Salta se uguale a zero) di riga 21 farà sì che nel primo caso (loc. 2 diversa da zero) venga stampato il messaggio di avvertimento (righe 22 - 28), altrimenti (loc.2 = 0) procede alla cancellazione della memoria e al reset del sistema (righe 30 - 42).

Poco da dire sulla sezione che si occupa di stampare sullo schermo il messaggio di riga 46, se non che sfrutta l'indirizzamento assoluto indicizzato (già visto sul numero scorso della rivista), nonchè la direttiva di assemblaggio .BYTE (vedi riquadro) per installare in memoria i codici Ascii di "NON RIPROVARCI" ed uno zero che segnala la fine della frase. Questa può essere modificata secondo il vostro (buon) gusto, tenendo presente che è possibile inserire qualunque codice Ascii, compresi quelli che controllano lo schermo, il cursore, eccetera.

Con un 147 prima dei caratteri alfabetici (seguito da una virgola!), per esempio, si otterrà una cancellazione dello schermo, e così via. Per la trasmissione dei caratteri al video, viene utilizzata la routine CHROUT del Kernal, argomento questo che verrà meglio trattato prossimamente.

E veniamo alla sezione più "cattiva", quel-

Il Macroassembler Commodore offre una miriade di funzioni che rendono semplice la programmazione in l.m.

Il Macroassembler (disponibile solo su disco) potete acquistarlo presso i rivenditori autorizzati Commodore oppure, per corrispondenza presso i nostri inserzionisti

la che si occupa di cancellare il programma Basic residente in memoria.

Per svolgere questo compito faremo uso di un nuovo tipo di indirizzamento, definito "indicizzato indiretto", o, per evitare confusioni, "indiretto postindicizzato". Esso sfrutta come indice il registro Y (obbligatoriamente!), il cui contenuto deve essere sommato all'indirizzo "puntato" da due locazioni di memoria, la prima delle quali è specificata nell'operando dell'istruzione Assembly. I due puntatori devono essere due locazioni di pagina zero, come vengono definite le prime 256 locazioni di memoria del C/64.

Per capire meglio, supponiamo che nella locazione $FB (decimale 251) sia contenuto uno zero, e nella successiva ($FC = decimale 252) il valore C0 (decimale 192). Considerando le due celle come puntatori, l'indirizzo in esse contenuto sarà $C000 (49152). Se ora immaginiamo che il registro Y sia stato caricato con $FF (255), ecco che un'istruzione tipo...

```
LDA ($FB),Y
```

...caricherà A con il contenuto della locazione esadecimale $C000 + $FF = $C0FF, ovvero, in decimale, 49152 + 255 = 49407.

Sulla base di queste considerazioni, ecco dunque come procede la seconda parte della nostra routine:

1) Prepara i due puntatori in pagina zero ponendo 0 in $FB, mentre in $FC viene copiato il contenuto della locazione 46 (BASIC + 3, vedi riga 04), che sappiamo essere il byte alto dell'indirizzo di fine programma Basic (righe 30 - 33).
2) Si prepara il registro Y con il valore $FF (riga 34).
3) Si azzerano tutte le locazioni che vanno dall'indirizzo di fine del programma Basic fino a $0800 (2048 = inizio Basic), decrementando opportunamente il byte alto del puntatore $FC ogni volta che il valore di Y, a sua volta decrementato, risulterà uguale a zero.
4) Si salta al reset di sistema, notoriamente situato in $FCE2 (64738).

Se si intende eliminare "l'avvertimento", basterà semplicemente sopprimere le istruzioni che lo riguardano (righe 16, 20-28, 44-47), sostituendo la label START presente negli operandi di riga 12 e 13 con il nuovo inizio dell'antilist, che corrisponderà all'etichetta CANCEL.

Inutile aggiungere che, per apportare queste (o altre) modifiche alla routine, il modo più semplice consiste nello sfruttare le facilitazioni offerte da un buon editor/assembler, e, per nostra fortuna, in questo settore Mamma Commodore non ci ha abbandonati al nostro destino.

UN MACROASSEMBLER AI NOSTRI ORDINI

Ogni assemblatore che si rispetti, ed il Macro Assembler Commodore non può che rientrare in questa categoria, dispone di un certo numero di istruzioni per così dire "riservate", in grado di facilitare ulteriormente la realizzazione di un programma in linguaggio macchina.

I comandi di cui stiamo parlando, che variano nei diversi package in commercio, vengono in genere definiti con il termine "pseudo-opcode" (opcode= codice operativo), in quanto non corrispondono a veri e propri codici riconoscibili dal microprocessore: pur essendo infatti manipolabili come qualunque altra istruzione Assembly, gli pseudo-opcode sono in realtà delle informazioni rivolte esclusivamente all'assembler, che si occuperà di applicarle utilizzando un collage di istruzioni "normali".

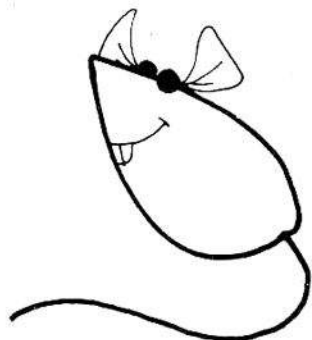
Per non andare troppo lontano, un tipico esempio è rappresentato dal segno "=", adoperato per assegnare un valore ad un simbolo, o ancora l'ormai noto asterisco seguito dal segno di uguaglianza ("*=") che indica al Macro Assembler da quale cella di memoria iniziare l'allocazione del programma.

Oltre ai due appena accennati, il Macro Assembler è in grado di riconoscere ben altre nove direttive speciali, tutte caratterizzate dalla presenza del carattere punto (.) come primo carattere, cui è dedicato un ampio settore del manuale in dotazione.

Lo pseudo-opcode di più frequente utilizzazione è senz'altro l'istruzione .BYTE, preposta ad inserire con facilità, nelle locazioni di memoria volute dall'utente, valori numerici in qualunque notazione o testi Ascii secondo modalità che risulteranno più evidenti considerando i seguenti esempi:

1) NUMERO .BYTE 255
2) NUMERI .BYTE $FF, 147, %0111
3) TESTO .BYTE 'COMMODORE'

Nei tre casi considerati, l'assembler riserverà lo spazio necessario ed inserirà i valori precisati dall'operando nelle locazioni a partire da quella specificata dalle etichette, come avviene per qualunque "vera" istruzione Assembly: quindi nell'esempio 1 verrà caricato il valore decimale 255 nella singola locazione identificata dalla label NUMERO, mentre nel caso 2 verrà riservato lo spazio di 3 locazioni, che conterranno, nell'ordine, quanto descritto dall'operando; questo può contenere più informazioni, purchè separate da una virgola, e, come si può notare, i valori possono essere espressi indifferentemente in notazione decimale, esadecimale, o binaria (nonchè in ottale, se può interessarvi!).

Nel terzo esempio, vediamo una delle più comode opzioni della direttiva .BYTE: adoperando il carattere ottenibile premendo i tasti Shift + 7 (apostrofo) come delimitatore, l'assembler terrà in considerazione il valore Ascii dei singoli caratteri, consentendo applicazioni come quella proposta in queste pagine.

TUTTI I TEMPI DEL VIDEO

Può capitare di dover scrivere, per realizzare effetti insoliti, alcune routine sincrone al raster register

di **Lorenzo Emilitri**

La manipolazione del raster non finisce mai di stupire

Il sistema PAL, ovvero il sistema televisivo usato in Italia, prevede che il quadro video venga inviato, al cinescopio, 50 volte al secondo. Ciò significa che ogni quadro video distinto ha una durata complessiva di 20 millisecondi.

Il C/64, inoltre, divide il video in 312 linee orizzontali, quindi la durata di ogni singola riga è di (circa) 64 microsecondi.

Da ciò si potrebbe dedurre che una routine di 64 cicli sia perfettamente sincrona con lo schermo.

Per verificare questa affermazione è stata progettata una routine che legge, ogni 64 cicli, il valore della locazione $D012 (byte basso del raster register) e lo deposita in un buffer a partire da $C100.

Questa routine, in una prima versione, avrebbe dovuto produrre una serie di numeri consecutivi a partire da $C100 fino a $C1FF.

Effettuando prove, tuttavia, una volta mandata in esecuzione, la routine di 64 cicli "saltava" alcuni numeri di riga e non li immagazzinava nel buffer da $C100.

Apportando modifiche alla lunghezza stessa della routine (prima in 65 cicli e poi in 63) quest'ultimo valore dimostrava di essere in perfetto sincronismo con il quadro video.

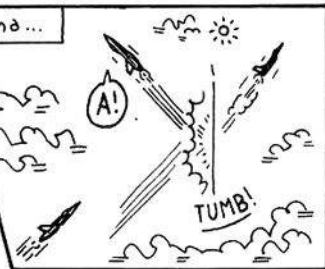
Da ciò si deduce che la lunghezza di un impulso di clock del C/64 non è, come affermano alcuni testi, di 1 microsecondo esatto ma, in realtà, di 64 / 63 = 1.015 microsecondi. La vera frequenza di clock non è quindi di 1 mhz, ma di 985 khz; tale differenza non è affatto trascurabile, come andremo a dimostrare.

Osservando inoltre lo schema elettrico del C/64 si può notare che esiste una frequenza (il DOT CLOCK) utilizzata dal VIC che è di 7.88 mhz, ovvero esattamente 8 volte la frequenza di clock sopra riportata.

Da tutto questo discorso si potrebbe quindi di pensare che una routine di 63 cicli sia perfettamente sincrona al raster register.

N. Linea	1' valore	1' val. 01
0 o 8	0E	3' posto
1 o 9	0E	2' posto
2 o A	06	9' posto
3 o B	06	8' posto
4 o C	0F	7' posto
5 o D	0F	6' posto
6 o E	0E o 0F	5' posto
7 o F	0E o 0F	4' posto

Ancora una volta, però, andiamo incontro ad una cocente delusione: una routine del genere non porta al risultato voluto.

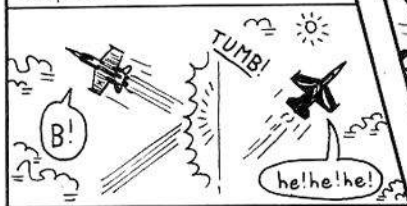
Infatti ogni otto righe dello schermo visibile, di lunghezza 63 cicli, c'è una linea in cui si possono utilizzare solo 35 cicli, in quanto il microprocessore viene bloccato per gli altri 28 cicli dal VIC II che deve (probabilmente) accedere alla memoria per visualizzare i caratteri o la pagina grafica. Ciò non acca-

```
;  *******************************
;  * misura lunghezza raster v2.1 *
;  * lunghezza routine:  63 cicli *
;  *******************************
;   by lorenzo emilitri 10.09.1988
;
        *=$c000        ;inizio a 49152
;
        sei            ;disab. irq
        lda blank      ;blanking di
        and #239       ;schermo per
        sta blank      ;disab. nmi
wait    lda raster     ;aspetta la
        bne wait       ;riga 0
        ldy #00        ;contatore = 0
loop1   lda raster     ;carica raster
        sta buffer,y   ;e immagazzinalo
        ldx #09        ;loop per
loop2   dex            ;ritardo
        bne loop2      ;di 46 cicli...
        bit $00        ;piu' altri 3
        iny            ;incr. contat.
        bne loop1      ;se non e' zero
                       ;salta a loop1
        lda blank      ;schermo visi-
        ora #16        ;bile (abilita
        sta blank      ;nmi)
        cli            ;abilita irq
        rts            ;torna al basic
;
;  labels:
;
blank  =$d011
raster =$d012
buffer =$c100
```

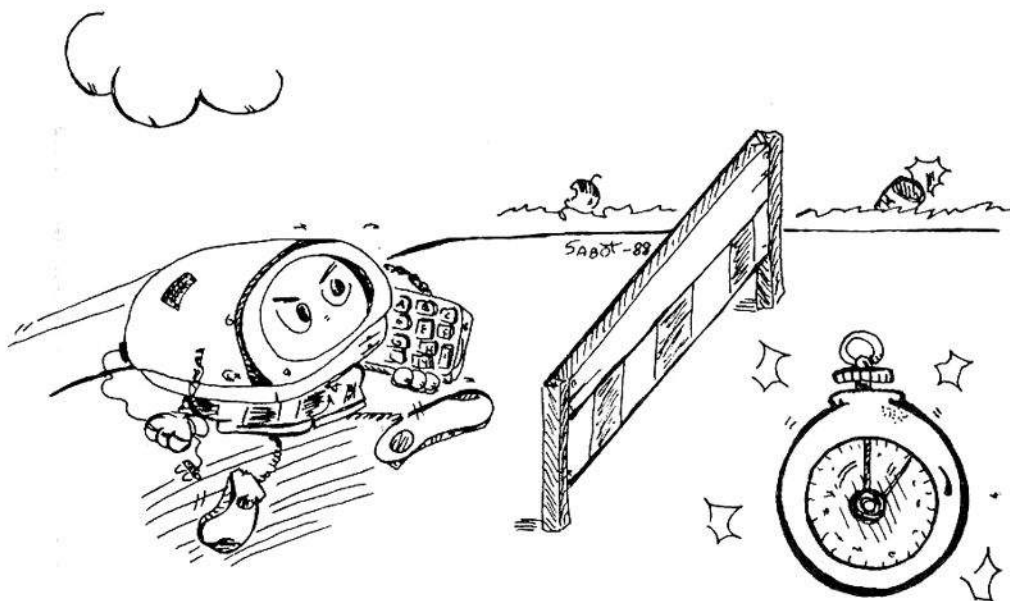
de se, tramite il bit 4 della locazione $D011, abbiamo abilitato il blanking del video.

Le linee in questione sono quelle comprese tra la 50 e la 250 il cui valore esadecimale termina con la cifra "3" o con la cifra "B"; la prima è la 51 e l' ultima ad essere visualizzata è la 243.

Da tutto il discorso precedente si può quindi dedurre che un quadro video completo dura 18956 cicli (se il video è visibile) oppure 19656 cicli se il modo blanking è attivo; ciò conferma il fatto che il blanking di schermo, in effetti, accelera la velocità di elaborazione di circa il 4 per cento.

Un discorso a parte meritano gli interrupt IRQ che si susseguono (in condizioni normali) ogni 16422 cicli e durano per un minimo di 218 cicli l'uno. Inoltre spesso capita che due IRQ avvengano nello stesso quadro video con una perdita totale, nel migliore dei casi, di ben 436 cicli.

```
;  **************************
;  * inclina caratteri  v2.0 *
;  **************************
;  by lorenzo emilitri 12.9.88
;
     *=$c000        ;inizio: 49152
; routine preparatoria
     sei            ;disab. irq
     lda #<main     ;fai puntare
     sta $0314      ;il vett. di irq
     lda #>main     ;a main
     sta $0315      ;
     lda #$3f       ;forza il raster
     sta raster     ;interrupt a $3f
     lda hirast     ;azzera il msb
     and #$7f       ;del raster
     sta hirast     ;
     lda #$00       ;disabilita il
     sta timer      ;timer
     lda #$01       ;abilita i raster
     sta vicirq     ;interrupts
     cli            ;riabilita irq
     rts            ;ed esce
; routine che gira sotto interrupt
main lda #$01       ;conferma raster
     sta vicflg     ;interrupt
     ldx #$00       ;azzera contatore
loop lda valori,x   ;carica il valore
                    ;di shift laterale
     ldy tempi,x    ;carica il ritardo
rit  dey            ;ciclo di
     bne rit        ;ritardo
     sta vctrl      ;imposta il valore
                    ;per shift later.
     nop            ;breve ritardo
     nop            ;(4 cicli)
     inx            ;incr. contatore
     cpx #$40       ;se non e' $40
     bne loop       ;salta a loop
     jmp irq        ;salta alla normale
                    ;routine di irq
;
; labels:
raster =$d012
hirast =$d011
timer  =$dc0e
vicirq =$d01a
vicflg =$d019
vctrl  =$d016
irq    =$ea31
valori =$c140
tempi  =$c100
```

La meticolosa determinazione del numero di cicli macchina conduce alla creazione di routine incredibili

```
; ******************************
; * strisce  colorate sincrone *
; * al raster register * v 2.0 *
; ******************************
; by lorenzo emilitri 09.09.1988
;
        *=$c000
; routine preparatoria
        sei
        lda #<main   ;punta l'interrupt
        sta $0314    ;a main
        lda #>main   ;
        sta $0315    ;
        lda #$3f     ;raster interr. alla
        sta raster   ;riga $3f
        lda hirast   ;resetta msb
        and #$7f     ;del raster register
        sta hirast   ;
        lda #$00     ;disab. timer a
        sta timer    ;del cia #
        lda #$01     ;abilita il raster
        sta vicirq   ;interrupt
        cli          ;riabilita interrupts
        lda #$01     ;posiz. 1'striscia: 1
        sta pos1     ;
        lda #$0c     ;posiz. 2'striscia:12
        sta pos2     ;
        lda #$18     ;posiz. 3'striscia:24
        sta pos3     ;
        lda #$00     ;direz. di tutte le
        sta dir1     ;strisce: salita
        sta dir2     ;
        sta dir3     ;
        rts          ;torna al basic
; routine interrupt aggiuntiva
main    lda #$01     ;conferma raster
        sta vicflg   ;interrupt
        ldx #$00     ;contatore = 0
lp1     lda colori,x;carica il colore
        ldy tempi,x ;carica il val. di ri-
lp2     dey          ;tardo e loop
        bne lp2      ;
        sta bordo    ;metti il colore nel
        sta video    ;bordo e nel video
        inx          ;increm. il contatore
        cpx #$37     ;se non e' 55 salta
        bne lp1      ;a lp1
        jsr rep      ;salta a subroutine
        jmp irq      ;salta a routine irq
; subroutine rep:
; chiama la subroutine move
; 3 volte
rep     ldx #$00     ;contatore = 0
lp3     jsr move     ;esegue subrout. move
        inx          ;increm. contatore
        cpx #$03     ;se non e' 3 salta
        bne lp3      ;a lp3
        rts          ;se no esce
; routine move:
; muove le tre righe in alto
; o in basso
move    lda dir1,x   ;carica direzione
        beq decr     ;se e' 0 : salita
        lda pos1,x   ;carica posiz. attuale
        tay          ;
        iny          ;incrementala di 1
        cpy #$30     ;se e' 48
        beq top      ;salta a top
        tya          ;
        sta pos1,x   ;salva la nuova pos.
        jsr draw     ;disegna una striscia
        rts          ;ed esce
top     lda #$00     ;nuova direzione:
        sta dir1,x   ;salita
        rts          ;esce
decr    lda pos1,x   ;carica posiz. attuale
        tay          ;
        dey          ;decrementala di 1
        beq bott     ;se e' 0 vai a bott
        tya          ;
        sta pos1,x   ;salva la nuova pos.
        jsr draw     ;disegna una striscia
        rts          ;ed esce
bott    sta dir1,x   ;nuova direz.:discesa
        rts          ;ed esce
; routine draw:
; trasferisci i dati-colore
; nella zona opportuna
draw    txa          ;salva il contatore
        pha          ;nello stack
        ldx #$00     ;azzera il contatore
lp4     lda data,x   ;trasferisce 8 bytes
        sta colori,y;da 'data' alla zona
        iny          ;opportuna di 'colori'
        inx          ;
        cpx #$08     ;
        bne lp4      ;
        pla          ;riprende vecchio
        tax          ;contatore dallo stack
        rts          ;ed esce
;
; labels:
raster  =$d012
hirast  =$d011
timer   =$dc0e
vicirq  =$d01a
vicflg  =$d019
bordo   =$d020
video   =$d021
irq     =$ea31
tempi   =$c100
colori  =$c150
data    =$c140
pos1    =$c148
pos2    =$c149
pos3    =$c14a
dir1    =$c14b
dir2    =$c14c
dir3    =$c14d
```

IL PRIMO PROGRAMMA

Dopo tutta questa noiosa (ma vitale) teoria, siamo ora pronti per scrivere una routine sincrona con lo schermo.

Un'applicazione tipica, di facile soluzione, è l'inclinazione a destra o a sinistra dei caratteri sul video.

Come si può vedere dal disassemblato 2, l'operazione risulta molto semplice: è sufficiente avere in memoria due tabelle: la prima per i valori di ritardo, l'altra per i valori da impostare in $D016 in modo da ottenere lo scrolling laterale.

Per inserire in memoria la tabella dei tempi è stata usata una subroutine Basic che può essere utilizzata in qualsiasi programma per ridurre il numero delle linee DATA nel programma stesso.

Se volete quindi utilizzare la routine proposta nel programma 2 in qualche vostro programma, e avete bisogno dell'area $C000 - $D000 per altri scopi, è sufficiente riassemblare il disassemblato 2 in un'altra parte della memoria.

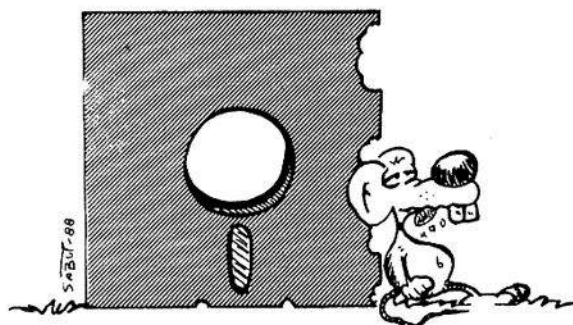
Contemporaneamente dovrete ricordare di cambiare gli indirizzi d'inizio della tabella dei tempi e di quella dei valori di $D016, e quindi modificare opportunamente i valori di X2 (inizio) e X3 (fine) per chiamare la routine.

Se, inoltre, volete alterare anche la posizione del raster interrupt cambiando il valore da forzare in $D012 dovete ricordarvi di modificare anche i valori di X0 e X1 rispettivamente con il primo ed il secondo valore della tabella 1, dove "N. Linea" indica la cifra meno significativa (in esadecimale) del nuovo raster interrupt.

I valori della tabella 1 sono studiati apposta per le due routine pubblicate o per qualsiasi vostra routine che sia realizzata sul loro modello con la stessa lunghezza (misurata rigorosamente in cicli macchina).

Essi indicano, rispettivamente, il primo valore della tabella dei ritardi, diverso dagli altri per evitare che il cambio di colori avvenga in una zona visibile, e il primo valore di ritardo 01 (= righe di 35 cicli) nella tabella.

Miscelare tra loro vari effetti grafici rappresenta una "sfida" continua alle potenzialità del sistema

Nei giorni successivi, le missioni si susseguono incessantemente. Quando mancano solo due giorni alla fine del corso, in testa al trofeo "Tom Bang" restano il nostro Giovedini e il tenente Iceman, che continua a barare sulla classifica...

L'inclinazione dei caratteri e lo scorrimento continuo di strisce sono solo due degli effetti ottenibili

Tutti questi valori "01" sono, come si è visto nella parte teorica, essenziali per la corretta temporizzazione e vanno intervallati, tra di loro, con 7 byte di valore 08 (= righe di 63 cicli).

Come potete vedere dal suo disassemblato, la routine per inclinare i caratteri sullo schermo utilizza il processore in modo esclusivo mentre è in esecuzione e porta un certo rallentamento in tutte le altre operazioni. Perciò è sconsigliabile utilizzarla in programmi che devono svolgere calcoli matematici complessi o altre operazioni di una certa lunghezza.

IL SECONDO PROGRAMMA

Il secondo programma, visibile nel disassemblato 3, muove tre strisce colorate orizzontali, su e giù per lo schermo, realizzando un effetto suggestivo.

Ogni striscia è alta 8 pixel, ognuno di colore diverso e liberamente impostabile dall'utente, con l'unica limitazione che il primo e l'ultimo dovrebbero essere dello stesso colore per evitare (brutti) effetti collaterali nella zona in cui le strisce si muovono.

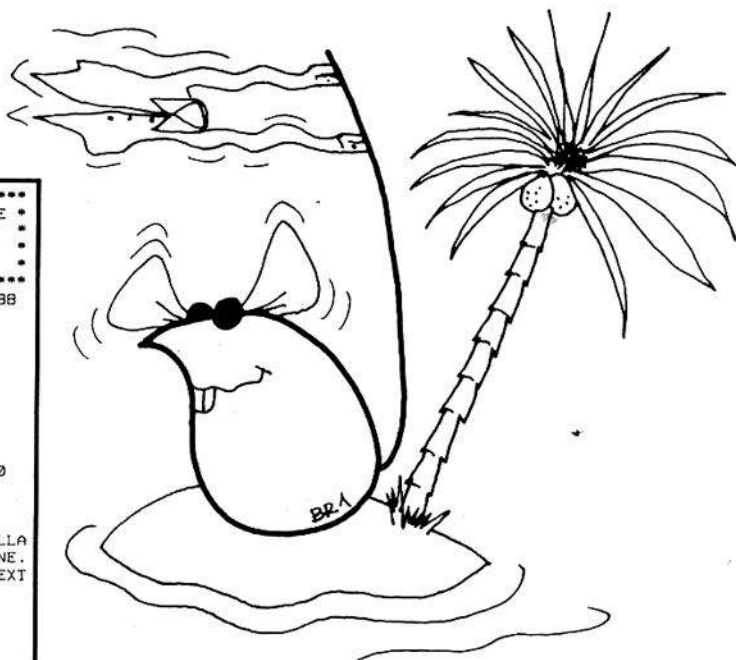
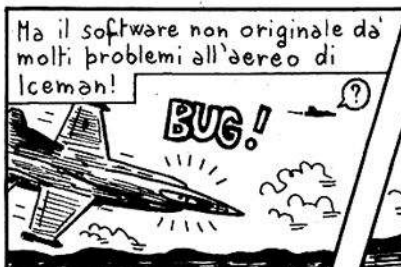
Anche questa routine può essere liberamente inserita in propri programmi per dare un tocco di colore allo schermo. Per rilocarla, o per spostare le strisce in un'altra area di schermo, valgono le stesse considerazioni della routine precedente.

```
100 REM ******************************
110 REM * INCLINA CARATTERI  V 2.0 *
120 REM *  -- CARICATORE BASIC --  *
130 REM ******************************
140 REM BY LORENZO EMILITRI
150 REM
160 FOR A = 0 TO 64 : READ B : C - C+B
170 POKE 49152 + A, B: NEXT
180 IF C <> 8098 THEN 480
190 X0-14 : X1-2 : X2=49408 : X3-49471
200 GOSUB 290
210 FOR A - 0 TO 7
220 FOR B - 0 TO 7
230 POKE 49472+A*8+B,B+200
240 NEXT B,A
250 SYS 49152
260 POKE 53280,0 : POKE 53281,0
270 PRINT CHR$(147) CHR$(30) TAB(12);
275 PRINT "ROUTINE ATTIVATA"
280 END
290 REM ROUTINE CHE PREPARA*LA TABELLA
300 REM DEI TEMPI USATA DALLA ROUTINE.
310 FOR A - X2 TO X3 : POKE A,8 : NEXT
320 POKE X2,X0
330 FOR A = X2+X1-1 TO X3 STEP 8
340 POKE A,1 : NEXT
350 RETURN
360 :
370 DATA 120,169,036,141,020,003,169
380 DATA 192,141,021,003,169,049,141
390 DATA 018,208,173,017,208,041,127
400 DATA 141,017,208,169,000,141,014
410 DATA 220,169,001,141,026,208,088
420 DATA 096,169,001,141,025,208,162
430 DATA 000,189,064,193,188,000,193
440 DATA 136,208,253,141,022,208,234
450 DATA 234,232,224,064,208,237,076
460 DATA 049,234
470 END
480 PRINT "ERRORE NEI DATA!" :END
```

PER FINIRE

Le routine qui presentate vanno considerate dal lettore non solo come programmi da digitare, ma soprattutto come laboratorio in cui sperimentare le conoscenze apprese in questo articolo.

I più esperti possono unire le due routine in modo da avere sul video contemporaneamente le strisce in movimento e i caratteri inclinati; oppure le strisce in movimento potrebbero muoversi sopra altre strisce ferme in altri colori: il limite agli effetti realizzabili è rappresentato solo dalla vostra fantasia che, siamo sicuri, non mancherà di produrre effetti di notevole complessità.

```
100 REM ********************************
110 REM * STRISCE COLORATE  SINCRONE *
120 REM *      AL RASTER REGISTER        *
130 REM *                                *
140 REM *       CARICATORE BASIC         *
150 REM ********************************
160 REM BY LORENZO EMILITRI 10.7.1988
170 REM
180 FOR A = 0 TO 171 : READ B
190 POKE 49152+A,B : C=C+B
200 NEXT A
210 IF C <> 22125 THEN 650
220 X0 = 14 : X1 = 4 : X2 = 49408
225 X3 = 49471 : GOSUB 310
230 FOR A = 0 TO 7 : READ B
240 POKE 49472+A,B : NEXT
250 FOR A = 0 TO 63 : POKE 49488+A,0
260 NEXT
270 SYS 49152
280 END
290 REM ROUTINE CHE PREPARA LA TABELLA
300 REM DEI TEMPI USATA DALLA ROUTINE.
310 FOR A = X2 TO X3 : POKE A,8 : NEXT
320 POKE X2,X0
330 FOR A = X2+X1-1 TO X3 STEP 8
340 POKE A,1 : NEXT
350 RETURN
360 :
370 DATA 120,169,062,141,020,003,169
380 DATA 192,141,021,003,169,063,141
390 DATA 018,208,173,017,208,041,127
400 DATA 141,017,208,169,000,141,014
410 DATA 220,169,001,141,026,208,088
420 DATA 169,001,141,072,193,169,012
430 DATA 141,073,193,169,024,141,074
440 DATA 193,169,000,141,075,193,141
450 DATA 076,193,141,077,193,096,169
460 DATA 001,141,025,208,162,000,189
470 DATA 080,193,188,000,193,136,208
480 DATA 253,141,032,208,141,033,208
490 DATA 232,224,055,208,236,032,095
500 DATA 192,076,049,234,162,000,032
510 DATA 106,192,232,224,003,208,248
520 DATA 096,189,075,193,240,023,189
530 DATA 072,193,168,200,192,048,240
540 DATA 008,152,157,072,193,032,153
550 DATA 192,096,169,000,157,075,193
560 DATA 096,189,072,193,168,136,240
570 DATA 008,152,157,072,193,032,153
580 DATA 192,096,157,075,193,096,138
590 DATA 072,162,000,189,064,193,153
600 DATA 080,193,200,232,224,008,208
610 DATA 244,104,170,096
620 REM VALORI DEI COLORI
630 DATA 00,08,02,07,05,04,05,00
640 END
650 PRINT"ERRORE NEI DATA!" :END
```

```
100 REM ********************************
110 REM * TROVA LUNGHEZZA RASTER V2.2 *
120 REM *    -- CARICATORE  BASIC --    *
130 REM ********************************
140 REM BY LORENZO EMILITRI
150 :
160 FOR A = 0 TO 41 : READ B : C = C+B
170 POKE 49152+A,B : NEXT
180 IF C <> 5418 THEN 350
190 SYS 49152
200 POKE 53280,0 : POKE 53281,0
210 PRINT CHR$(147) TAB(5);
215 PRINT "VALORI DA $C100 A $C17F":PRINT
220 FOR A = 0 TO 15
230 FOR B = 0 TO 7
240 A$=RIGHT$("    "+STR$(PEEK(49408+A*8+B)),4)
250 PRINT A$; : NEXT B : PRINT : NEXT A
260 END
270 :
280 DATA 120,173,017,208,041,239,141
290 DATA 017,208,173,018,208,208,251
300 DATA 160,000,173,018,208,153,000
310 DATA 193,162,009,202,208,253,036
320 DATA 000,200,208,240,173,017,208
330 DATA 009,016,141,017,208,088,096
340 END
350 PRINT "ERRORE NEI DATA!" :END
```

IN NOME DEI KERNAL

Per "esplorare" il drive è bene effettuare alcuni studi particolari sulle principali routine sia del computer che del drive stesso

di **Alessandro Diano**

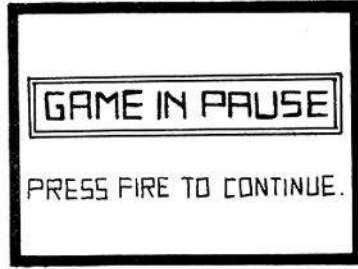
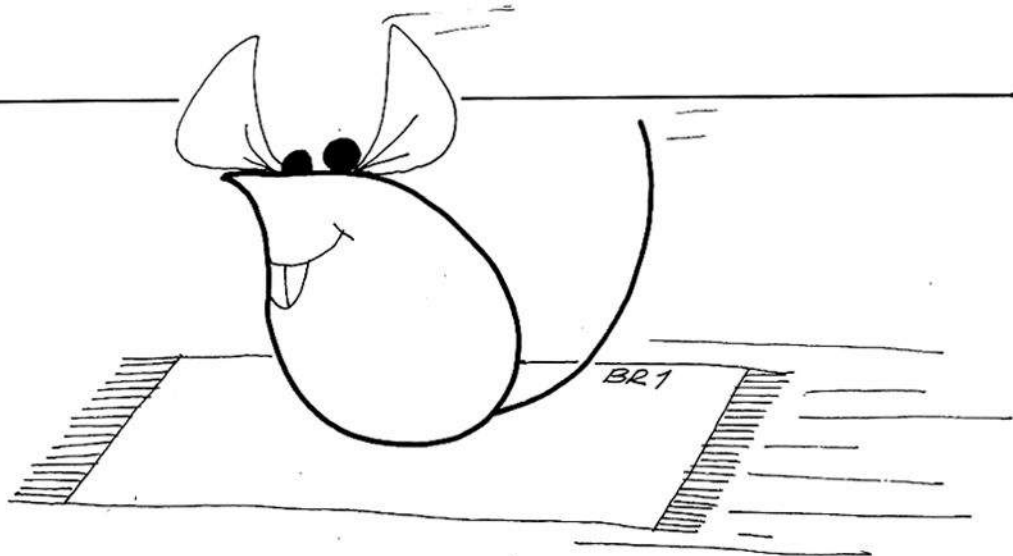
Il Kernal rappresenta il gruppo di routine l.m. più importanti del sistema

Quando si programma in L.M. qualche routine di I/O sul proprio computer, solitamente vi è la tendenza a servirsi di alcuni JMP del Kernal, piuttosto che di altri, anche perchè, usando le routine già conosciute, si incontreranno sicuramente meno problemi.

Nella Hit Parade del "Kernal for I/O", infatti, risultano essere maggiormente gettonate le routine di...

SETNAM ($FFBD)
per la specifica dei caratteri da inviare dopo la OPEN (tipo Initialize, il cancelletto # per l'accesso diretto, etc.),
SETLFS ($FFBA)
per i parametri del file, numero di periferica ed eventuale indirizzo secondario,
OPEN ($FFC0)
per l'omonima operazione, cioè per l'apertura vera e propria del file,
CHKOUT ($FFC9)
per preparare l'invio in uscita dei dati verso il file aperto (routine usata dal S.O. per il comando basic CMD x, con x = numero logico del file),
CHROUT ($FFD2)
sicuramente la prima in classifica in ordine d'uso e serve ad inviare i dati sul corrente canale aperto in uscita (dalla CHKOUT),
CHKIN ($FFC6)
che prepara un file in ingresso (l'inverso di CHKOUT),
CHRIN ($FFCF)
per ricevere i dati in arrivo (o, meglio, "prelevarli" dalla periferica esterna),
CLOSE ($FFC3)
che chiude il canale di comunicazione al termine delle operazioni di ingresso / uscita dei dati da e per il dispositivo esterno che, nel caso che stiamo esaminando, è un disk drive.

Orbene si nota che tali routine fanno tutte parte del kernal (sono riconoscibili dalle prime due cifre esadecimali "FF" dell'indirizzo di chiamata della routine stessa) ed impiegano altri JMP $FFxx della tavola dei salti della pagina numero 255 del nostro computer, sia esso VIC 20, C/64, C/16, Plus/4 oppure C/128; tali "altre" routine sono tra le meno usate di tutto il kernal, proprio per la loro caratteristica di interagire in maniera diretta con la R.O.M. delle unità a disco, obbligando, in tal modo, a conoscere direttamente la gestione dei dati da parte di un S.O. che NON è quello dell'elaboratore centrale.

Queste "sconosciute" sono in otto in totale: la prima metà di esse gestisce il transito dei dati da drive a computer mentre le altre quattro si occupano dell'invio dei dati verso

I COMANDI PER IL 1571

Il disk drive 1571 è, sostanzialmente, una copia migliorata del già ben noto fratello minore 1541; possiede tuttavia alcuni nuovi comandi dedicati, non riconosciuti dal 1541:

OPEN 1, 8, 15, "U0>M0": CLOSE 1
Viene attivata la modalità operativa a singola faccia, propria del 1541.

OPEN 1, 8, 15, "U0>M1": CLOSE 1
Viene attivata la modalità operativa a doppia faccia, propria del 1571.

OPEN 1, 8, 15, "U0>H0": CLOSE 1
Seleziona la testina numero zero del disk drive (cioè la prima).

OPEN 1, 8, 15, "U0>H1": CLOSE 1
Seleziona la testina numero uno del disk drive (cioè la seconda); tale comando, ovviamente, è impiegabile solo ed esclusivamente nella modalità 1571 a doppia faccia.

OPEN 1, 8, 15, "U0>" + CHR$(x): CLOSE 1
Sostituendo "x" con un numero compreso tra 4 e 30 (generalmente viene scelto il nove), si cambia l'originale numero di periferica (normalmente otto) con il valore "x".

L'altra grossa novità del 1571 è rappresentata dal cosiddetto "Burst Command Instruction Set" (B.C.I.S.) cioè il set di istruzioni del comando burst con il quale sono possibili formattazioni, caricamenti e registrazioni in vari formati ad una velocità superiore rispetto a quella normale.

AGGIORNAMENTI R.O.M. 1541

E' appurato che esistono in commercio almeno due differenti sistemi operativi per 1541; in linea di massima i nuovi sono riconoscibili dalla "levetta" posta sul frontale in luogo del vecchio sportellino ma, a scanso d'equivoci, ecco le modifiche presenti nella nuova versione di S.O.

Nelle locazioni da 65296 ($FF10) a 65326 ($FF2E), anzichè l'originario valore di riempimento 170 ($AA), sono state aggiunte le due routine riportate nel disassemblato n. 4.

La prima aggiorna la routine di reset, mentre la seconda integra la parte che si occupa della lettura del bus seriale.

Tali routine sono state collegate al resto del S.O. con due JMP (prima non presenti) posti a $EAA4 ed a $E9DC come indicato nella tabella n. 3.

Ciò per consentire (via software) di distinguere tra i due S.O.; è già noto, infatti, che la R.O.M. 1541 si distingue da quella del 1571 valutando il contenuto della locazione 58822 ($E5C6) che vale, rispettivamente, 52 ($34) e 55 ($37).

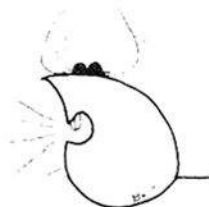
Prima di attivare una qualsiasi routine del Kernal è indispensabile settare vari parametri

la periferica esterna; al primo gruppo appartengono le routine...

TALK ($FFB4)
per comandare, ad una periferica collegata tramite il bus seriale, l'invio dei dati richiesti,

TKSA ($FF96)
per inviare l'indirizzo secondario ad un dispositivo esterno al quale è stato richiesto di inviare dei dati,

ACPTR ($FFA5)
per ricevere il carattere proveniente dal bus seriale,

UNTLK ($FFAB) per far terminare l'invio dei dati da parte della periferica esterna.

Il secondo gruppo è composto dalle routine...

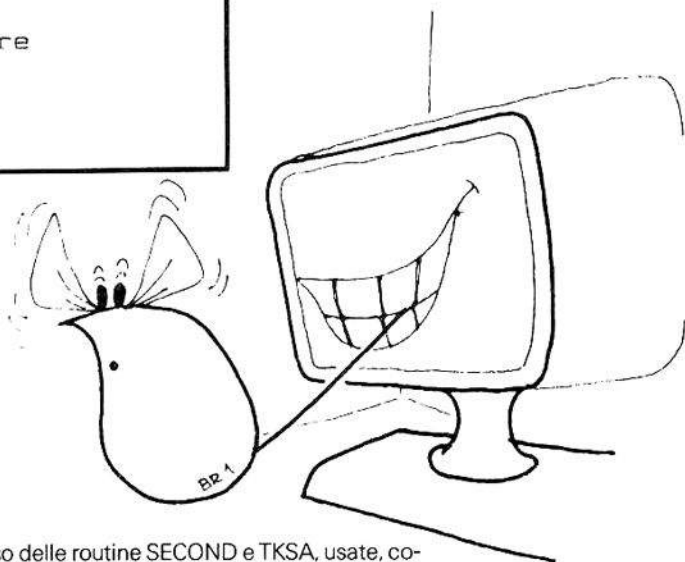
LISTEN ($FFB1)
per preparare il device a ricevere i dati provenienti dal computer,

VALORE DEC.	VALORE ESA.	OPERAZIONE
240	$F0	Apre un file
96	$60	Riapre un file
224	$E0	Chiude un file

Tabella n.1

```
LDA #$08    Periferica numero 8
JSR LISTEN
LDA #$F2    Indirizzo secondario 2 + $F0
JSR SECOND
LDA #$23    Carattere "#"
JSR CIOUT   Invia il carattere
JSR UNLSN

Disassemblato n.1
```

SECOND ($FF93)
per l'invio dell'indirizzo secondario dopo il LISTEN,
CIOUT ($FFA8)
per inviare un dato (in pratica otto bit, cioè un byte) attraverso, l'ormai famoso, bus seriale,
UNLSN ($FFAE)
che invia alla periferica esterna un segnale di "Fine dei dati" che la lasci nuovamente libera per nuove operazioni.

Su alcuni manuali (tipo la "Guida di riferimento per il programmatore" della Commodore), è presente un'accurata spiegazione sui vari registri da caricare prima delle chiamate a tali subroutine ma, purtroppo, manca un'importante specificazione sull'uso delle routine SECOND e TKSA, usate, come già visto, per l'invio dell'indirizzo secondario dopo, rispettivamente, il LISTEN (= ascolta) ed il TALK (=parla). Si tratta di alcuni

```
LDA #$08    Periferica numero 8
JSR LISTEN  (oppure TALK secondo i casi)
LDA #$62    Indirizzo secondario 2 + $60
JSR SECOND  (oppure TKSA se in TALK)
LDA #CHAR   (solo se in LISTEN)
JSR CIOUT   (oppure ACPTR se in TALK)
JSR UNLSN   (oppure UNTLK se in TALK)

Disassemblato n.2
```

```
LDA #$08    Periferica numero 8
JSR LISTEN
LDA #$E2    Indirizzo secondario 2 + $E0
JSR SECOND
JSR UNLSN

Disassemblato n.3
```

Notevole è l'importanza, nel computer e nel drive, delle routine contenute nelle ROM

codici "standard", propri dei disk drive, riassunti nella tabella n. 1 in cui vengono riportati i valori CORRETTI con i quali effettuare un OR con l'indirizzo secondario da inviare; se (come di solito accade) quest'ultimo è minore di 127 ($7F) si può invece eseguire una banalissima somma. Nel disassemblato n.1 è riportato un esempio dell'equivalente dell'istruzione basic OPEN 2, 8, 2, "#" per l'accesso diretto al drive.

A questo punto, dopo aver aperto, analogamente, il canale 15 (OPEN 15, 8, 15) usando $FF come indirizzo secondario ($0F che è, appunto, uguale a 15 più il codice di OPEN $F0) ci si deve ricordare di terminare SEMPRE con un UNLSN tutti i LISTEN ancora aperti.

Successivamente, quando si vorrà riaprire la comunicazione (che, nota bene, NON è stata chiusa dall'UNLSN, ma solo sospesa!), basterà la sequenza riportata nel secondo disassemblato.

Da notare che l'unica differenza sostanziale è nell'indirizzo secondario (il quale identifica anche il file logico) che viene inviato: al valore nominale 2 viene stavolta (ed in

BIT	VALORE	CASSETTA	BUS-SERIALE
0	1	-	Timeout in scrittura
1	2	-	Timeout in lettura
2	4	Blocco corto	-
3	8	Blocco lungo	-
4	16	Errore in lettura	-
5	32	Errore checksum	-
6	64	Fine del file	Fine linea
7	128	Fine del nastro	Periferica non presente

Tabella n.2

```
$FF10 STX $1803   Porta A in uscita.
$FF13 LDA #$02    Bit di uscita "alto".
$FF15 STA $1800   Porta B in uscita.
$FF18 LDA #$1A    Direzione dati "out".
$FF1A STA #$1802  "Poka" nel registro B.
$FF1D JMP $EAA7   Prosegue il reset.
----- --- -----
$FF20 LDA $1800   Legge il bus seriale.
$FF23 AND #$01    Bit "Data in".
$FF25 BNE $FF20   Attende il byte dati.
$FF27 LDA #$01    Setta il "Timer
$FF29 STA $1805   interrupt".
$FF2C JMP $E9DF   Prosegue lettura bus.

$FF2F $AA .....   Valori di riempimento.
```

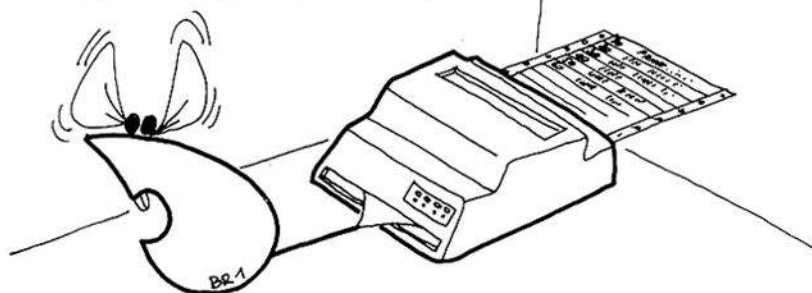
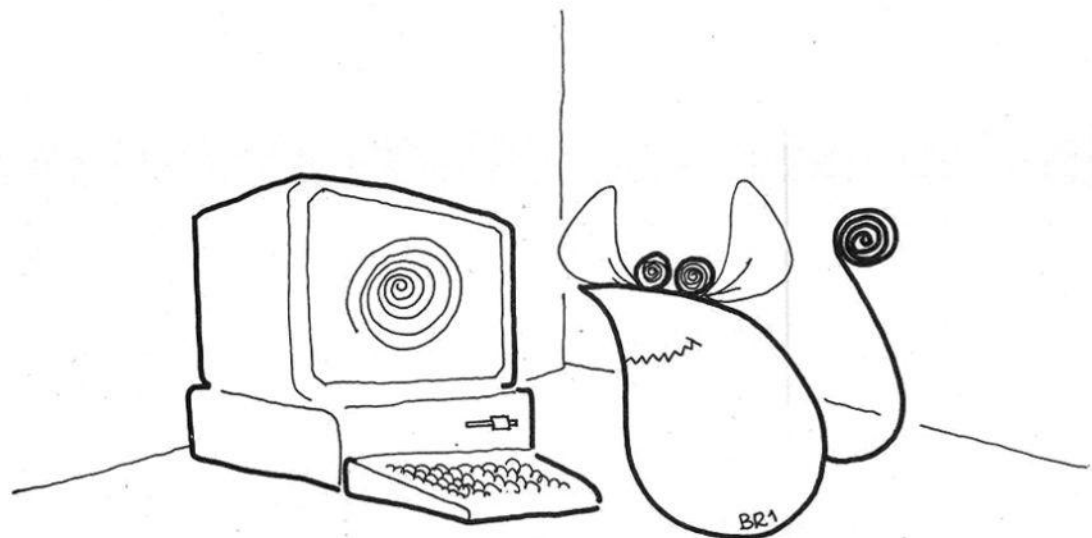
Disassemblato n.4

Vecchio S.O.	Aggiornamento
$EAA4 STX $1803	$EAA4 JMP $FF10
$E9DC STA $1805	$E9DC JMP $FF20

Tabella n.3

tutte le riaperture successive ad un UNLSN o ad un UNTLK) "sommato" il valore $60 con un OR...

$02 OR $60 = $62

...che, appunto, rappresenta il valore da passare alla routine di indirizzo secondario SECOND dopo un LISTEN, oppure TKSA dopo un TALK: le due routine, infatti, NON sono assolutamente interscambiabili (importante!).

Dovrebbe a questo punto essere chiaro (e se fosse altrimenti sarebbe opportuno rileggere l'articolo e rintracciare la malcomprensione) che quando si desidera chiudere definitivamente la comunicazione (l'equivalente del comando basic CLOSE) si dovrà far eseguire una routine del genere indicato nel disassemblato n. 3.

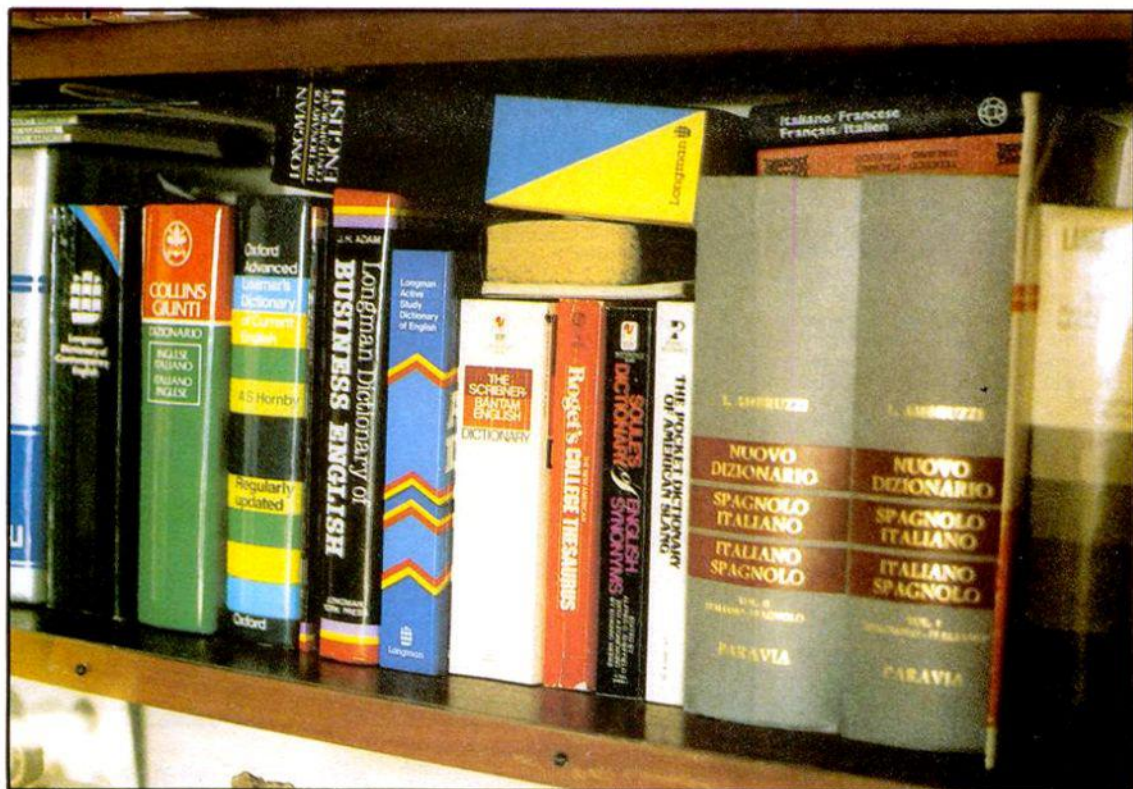
L'ultima raccomandazione, particolarmente importante usando queste routine, riguarda la locazione 144 ($90) che è l'analogo in L.M. della variabile basic riservata ST: è opportuno azzerarla subito prima di un'operazione I/O per poi leggerne il contenuto e gestire le indicazioni relative come da tabella n. 2.

Per quanto riguarda la gestione del bus seriale, il valore più importante è il 64: quando la locazione 144 contiene tale numero, significa che i dati che la periferica esterna doveva inviare sono terminati; per gli altri errori basta considerarne gli addendi, esempio: un valore di 66 ($42) rappresenta un segnale di fine linea (64) ed un errore di supero tempo in lettura (2), infatti 64 + 2 dà come risultato proprio il 66 trovato.

SIAMO ALFIN GIUNTI

...al termine dell'opera anche se, in realtà, ci si trova solo all'inizio: tali proposte, infatti, rappresentano un punto di partenza (e non certo d'arrivo) nell'esplorazione di una periferica "intelligente" come la nostra beneamata unità a dischi "made in C.B.M."; io ho cercato solo di "impulciare" qualche padiglione auricolare, per il resto basta armarsi di taccuino, drive e fantasia: buon lavoro!

E così, cari amici, con questa offerta di collaborazione, termina l'avventura di Primo Giovedini alla scuola del mitico Tom Bang, coronata dal più ampio successo!
Ed ora, arrivederci.

SYS 64738

LA LIBRERIA DI AMIGA

Come sfruttare le potenzialità di AmigaBasic utilizzando una nostra precedente idea ("Enciclopedia di routine in Basic") riveduta ed aggiornata per il potente computer della Commodore

di **Luigi Callegari**

Numerosi utenti possiedono (per svariati, dolorosi motivi sui quali non ci intratteniamo) il manuale Amigabasic solo in sintetico inglese, ed è quindi opportuno pensare anche a loro.

In AmigaBasic, d'altra parte, il concetto di sottoprogramma è nuovo rispetto al concetto di subroutine e pensiamo che molti gradiranno avere alcune precisazioni, sia per imparare ad usare al meglio le routine di libreria per Amigabasic (che volta per volta proporremo) nei loro programmi, sia per poterne scrivere di efficienti da soli e magari inviarcele per la pubblicazione.

NOTE LINGUISTICHE

In italiano i termini "subprogram" e "subroutine" si traducono sempre con "sottoprogrammi", ma nel manuale di Amigabasic ci si riferisce a cose completamente diverse.

Tanto per iniziare, le subroutine sono invocate con un GOSUB, mentre i subprogram sono attivati mediante CALL o, addirittura, direttamente per nome, costituendo così un vero e proprio strumento di estensione di Amigabasic. E' proprio per questo motivo, e per le caratteristiche che hanno i subprogram in Amigabasic, che preferiamo chiamarli "funzioni", come in Pascal o C (anche se non sono proprio uguali.).

Non si confondano, comunque, le "funzioni" (che usano le parole SUB, END - SUB, EXIT - SUB, CALL) con le "funzioni definite dall'utente" che usano DEF FN e FN e sono tutt'altra cosa, nota, peraltro, agli utenti del caro C/64.

UTILITA' DELLE FUNZIONI

I nomi di variabili usati all'interno di una funzione, differentemente da quanto av-

viene nei sottoprogrammi, sono considerati interni alla funzione stessa.

Ciò significa che si possono usare, internamente alle nostre funzioni, anche nomi di variabili già usati nel programma principale, certi di non avere interferenze con il programma chiamante. Le variabili di questo tipo sono dette "locali", proprio perchè i loro valori non possono essere alterati all'esterno della funzione.

Le funzioni di AmigaBasic non possono però essere ricorsive, ovvero richiamare se stesse. Inoltre, uscendo da una funzione e poi rientrandovi, le sue variabili locali riassumono i valori che avevano al momento dell'ultima "uscita" da essa, ovvero non vi è una "allocazione dinamica" con cancellazione fisica delle variabili al termine della funzione. Proprio per questo non è ammessa la ricorsione, altrimenti l'area di memoria per le variabili si riempirebbe in fretta.

All'interno di una funzione si può usare tutto ciò che è consentito in AmigaBasic, tranne due eccezioni:

1) Non è ammesso usare CLEAR, COMMON e DEF FN
2) Non è ammesso definire funzioni all'interno di altre funzioni.

COME COSTRUIRSI UNA LIBRERIA DI FUNZIONI

Quando si vuole scrivere una funzione di libreria si può procedere come segue:

1) "Pulire" il Basic con un NEW
2) Digitare la sola funzione desiderata.
3) Inserire, in testa alla funzione, qualche linea REM per specificare i parametri necessari, quali sono i lavori svolti dalla funzione e quali i valori restituiti. Tutto questo, ovviamente, servirà come pro-memoria per successivi studi della funzione stessa.
4) Salvare la routine in formato ASCII per permetterne la successiva fusione (MERGE) con altri programmi in futuro, usando un comando diretto del tipo:

```
SAVE "df0: Libreria / nomef", a
```

In questo caso si salverà una funzione chiamata "nomef" nella directory "Libreria" (che deve esistere, ovvero essere stata creata da CLI con un comando MAKEDIR) sul dischetto presente nel drive interno (df0:), appunto in formato ASCII (indicato dalla "a" finale).

Quando, in seguito, vorremo inserire una funzione di libreria in un nostro programma, useremo semplicemente, in modo diretto...

```
MERGE "df0: Libreria / nomef"
```

...e l'elaboratore inserirà, in coda al programma, la nostra funzione, consentendoci di non riscriverla tutte le volte che serve.

Un ovvio consiglio: per garantire la massima portatilità delle funzioni di libreria, evitare l'uso di variabili SHARED.

COME SI SCRIVE UNA FUNZIONE

Per definire una funzione si usa un blocco di linee del tipo:

```
SUB nome funzione (arg1, arg2...) STATIC
............
... (linee) ...
............
END SUB
```

Ciò che si scrive tra la linea che inizia con SUB e quella che termina con END SUB costituisce il "corpo" della funzione stessa, ovvero la definizione del gruppo di istruzioni da eseguire ogni volta che la funzione è invocata con il suo nome.

La parola STATIC è obbligatoria.

Opzionalmente si può usare, come seconda linea, una specifica SHARED per definire una o più variabili che non sono locali alla funzione, ma in comune con il programma principale. In questo caso, modificandole all'interno della funzione, esse rimarranno modificate anche all'esterno del programma e viceversa.

Ciò che segue il "nome funzione" è l'elenco ordinato dei cosiddetti "parametri formali" della funzione, ovvero delle variabili usate dal sottoprogramma in corrispondenza degli argomenti passanti dalla routine chiamante.

Infatti un programma può invocare una funzione chiamata, diciamo "Mariella", indifferentemente nei due modi:

```
CALL Mariella (a, b%, c$, d())
...oppure...
Mariella a, b%, c$, d()
```

In tal caso la funzione avrà una definizione di questo tipo:

```
SUB Mariella (t, f%, c$, d() STATIC
.....
.....
END SUB
```

Si noti che i nomi dei parametri formali non sono eguali a quelli passati dal programma chiamante, ma sono obbligatoriamente identici per tipo.

Anzi, Amigabasic è assai pignolo su questo argomento perchè, ad esempio, scrivendo in un programma...

```
....................
CALL Calc (t, 5)
....................
SUB  Calc (a, b) STATIC
....................
END SUB
```

...si incapperà in una segnalazione di errore. Infatti il chiamante cerca di passare un parametro intero (5) mentre la definizione di funzione prevede come argomento formale una variabile in singola precisione (tipo di default delle variabili assunto da Amigabasic senza caratteri postfissi di definizione come $, %, !, #7. Per avere, quindi, un programma funzionante, si scriverà...

```
CALL Calc (t, 5.0)
```

...che l'editor trasformerà subito nella equivalente (che potevamo digitare direttamente):

```
CALL Calc (t, 5!)
```

...oppure, ovviamente, definire...

```
SUB Calc (a, b%) STATIC
```

...a seconda dei nostri scopi.

PASSAGGIO DI PARAMETRI

I parametri possono essere passati alla funzione per valore o per riferimento, come spiegato sul manuale al paragrafo 6/7. Inoltre si possono passare espressioni e intere matrici.

1) Per riferimento; esempio:

```
a=3: b=2: c=3.14: CALL giaggio (a, b)
PRINT a, b, c
END
SUB giaggio (x, y) STATIC
x = x+1: y = y-2
FOR c=1 TO 1000: NEXT
END SUB
```

Questo programmino stamperà i numeri 4, 0 e 3.14. Infatti ogni riferimento all'interno della funzione alle variabili X e Y influisce direttamente sulle variabili A e B del programma principale, essendo stato fatto esplicito riferimento dal programma chiamante.

Si noti che la variabile C resta inalterata (3.14) sebbene all'interno della funzione sia stata usata una variabile "C" (per un ciclo fittizio), ma il valore 1000 che assume in uscita dalla funzione resta separato dal valore 3.14. Questo per chiarire il concetto prima esposto di variabili locali.

2) Per valore; esempio:

```
a=1: b=2: Poffarre ((a), b)
PRINT a, b: END

SUB Poffarre (p, q) STATIC
p = p + 4: q = q + 7
END SUB
```

Questo programma stamperà i valori 1 e 9 perchè la variabile A, nella chiamata, è racchiusa tra parentesi, quindi si intede che ne viene passato il valore. Perciò il riferimento a P nella funzione (somma quattro) non influisce sulla variabile A del programma chiamante. Non così si comporta la variabile B, poichè la sua "controfigura" nella funzione "Q" viene incrementata di sette, dunque anche essa viene modificata appropriatamente nel programma principale, essendo stata passata per riferimento, come visto anche prima.

PASSAGGIO DI MATRICI

```
DIM a(100), b(10, 12)
.........
CALL summat (a(), b())
..........
SUB summat (x(1), y(2)) STATIC
..........
END SUB
```

Il programma chiamante passa alla funzione il nome di due variabili, così espresse dalla coppia di parentesi poste dopo il nome. Si noti che nella definizione della funzione i parametri formali specificano quante dimensioni devono avere le matrici passate dal programma chiamante. Nel nostro caso la funzione "summat" si attende che il primo nome di matrice ne indichi una unidimensionale e la seconda una bidimensionale, rispettivamente con un 1 e con un 2 tra parentesi.

Tentando di passare matrici con dimensioni diverse, si incappa in una segnalazione di errore. Non si confonda questo parametro di specifica del numero di dimensioni con la specifica del numero di elementi della matrice.

Come spiega il manuale al paragrafo 6/9, il programmatore può usare, all'interno del corpo della funzione, le specifiche LBOUND e UBOUND per ricavare i limiti delle matrici e poter lavorare convenientemente su di esse.

PASSAGGIO DI ESPRESSIONI

In Amigabasic è lecito passare una espressione ad una funzione. Concettualmente l'operazione è identica a quando si passa un parametro per valore, racchiudendolo tra parentesi nella chiamata. Infatti il valore dell'espressione rimane accessibile alla funzione, ma ovviamente non viene alterato nulla nel programma chiamante.

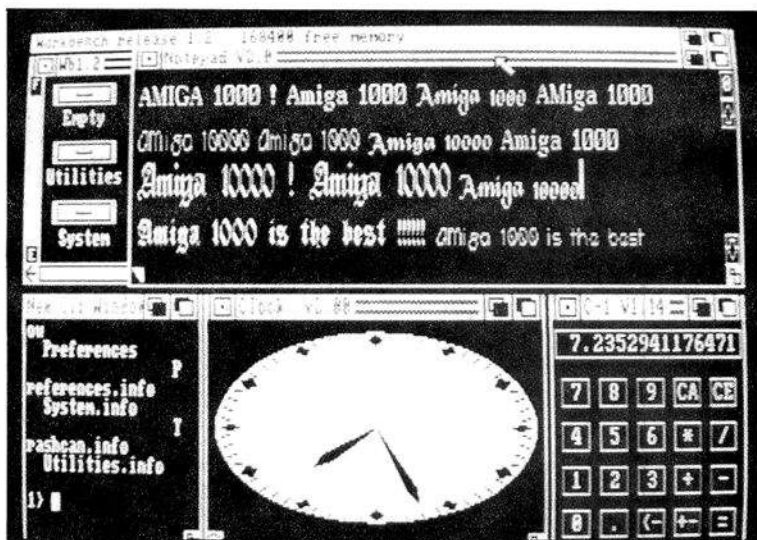
Ultima nota importantissima sull'uso delle funzioni, che il manuale trascura di spiegare nella sezione descrittiva, è che quando si desidera uscire da una funzione prima della sua fine si deve usare una specifica EXIT SUB.

Se non avete chiaro tutto quanto abbiamo detto, non preoccupatevi: sperimentate con i semplici esempi suggeriti e studiate le funzioni di libreria che pubblicheremo, volta per volta, su CCC.

SCRIVI ANCHE TU

Hai realizzato funzioni di libreria che ritieni siano "in linea" con quelle da noi proposte?

Telefona al nostro numero (02 / 84.67.348) per descriverle tenendo conto che, in caso affermativo, dovrai inviare le routine stesse e l'articolo relativo (scritto in Notepad) su dischetto al nostro indirizzo:

Systems Editoriale
Viale Famagosta 75
20142 Milano

LE TUE PRIME ROUTINE

Un gruppo di routine "universali", scritte in AmigaBasic, utili più spesso di quanto non possa sembrare a prima vista

di **Luigi Callegari**

GIUSTIFICAZIONI

Questa semplice funzione permette di ottenere una stringa "giustificata a destra", vale a dire allineata anche con il bordo destro.

Ciò si rivela utile, ad esempio, quando in un programma dobbiamo inserire lunghe frasi di testo, magari memorizzate sotto forma di DATA, e riportarle su video, o su stampante, perfettamente incolonnate sulla destra, in modo elegante come con un word processor.

Per invocare la funzione si usa una delle due linee qui di seguito suggerite:

CALL Justf (a$, l)

...oppure...

Justf a$, l

...in cui "A$" è la stringa da giustificare, mentre "L" è la lunghezza che si desidera ottenere. La stringa viene automaticamente portata alla lunghezza desiderata aggiungendo eventuali spazi tra le parole che la costituiscono.

Nella funzione, il ciclo WEND ... WHILE si chiude solo quando la stringa è della lunghezza voluta, mentre il ciclo FOR ... NEXT provvede a localizzare gli spazi tra le parole e ad inserirne altri per ottenere la lunghezza voluta.

Al "rientro" della routine al programma chiamante, la stringa A$ sarà della lunghezza desiderata. La funzione, tuttavia, non lavora se la stringa è già di lunghezza pari o superiore a quella voluta, oppure inferiore alla metà.

PINPUT

Questa routine costituisce l'adattamento di un programma già apparso per il C/64.

Permette, in sostanza, di stampare un messaggio su video e di ricevere, da tastiera, una stringa alfanumerica di risposta, escludendo o ammettendo caratteri specificati da un'altra stringa. Stiamo parlando, insomma, del famoso e classico "INPUT CONTROLLATO".

I formati per invocare PINPUT sono:

CALL PINPUT (a$, b$, c%)

...oppure...

PINPUT a$, b$, c%

...in cui "A$" specifica la stringa da stampare a video come richiesta di input. In uscita A$ conterrà la stringa di risposta. "B$" contiene un blocco di caratteri alfanumerici che saranno accettati, o respinti, se digitati da tastiera.

Ciò in funzione del valore di "C%", numero o variabile intera, che se vale zero indica di escludere i caratteri specificati in B$, se invece ha un valore diverso da zero indica che solo i caratteri specificati in B$ sono ammessi.

La pressione di un tasto durante l'input con un carattere non ammesso provoca un BEEP sonoro ed un lampeggio.

L'immissione della stringa termina quando si preme il tasto RETURN o ENTER. E' concesso l'uso del tasto DEL per cancellare l'ultimo carattere digitato.

QUICKSORT

Il Quicksort, inventato nel 1962 da Charles Hoare, è ancora oggi il più veloce algoritmo esistente per l'ordinamento sequenziale ("sort") di tabelle di dati.

Altri noti metodi di riordino (come il Sort selettivo, il Sort a bolle ed il Sort ad inserimento) richiedono, per N elementi, un tempo di calcolo proporzionale a t * (n * n), mentre il tempo per un Quicksort è proporzionale a t * (n * log (n)), dove "log" indica il logaritmo in base due. Ciò si risolve in una velocità di lavoro maggiore, tanto più sensibile quanto più numerosi sono i dati.

L'algoritmo funziona dividendo la matrice da riordinare in una parte "destra" che contiene gli elementi maggiori di un dato valore "A" ed una parte "sinistra" che contiene gli elementi inferiori. Il valore di A è arbitrario, ma è ottimale quando divide esattamente a metà la matrice da riordinare. Questa ripartizione viene eseguita sfruttando due puntatori ("i" e "j" nella nostra routine). Dapprima viene eseguita una scansione usando "i" per trovare un elemento maggiore di A, poi una scansione verso sinistra usando "j" per trovare un elemento inferiore. Trovandosi nelle sezioni errate della matrice, questi due elementi vengono scambiati (usando la funzione SWAP nella stessa routine).

Dopo il primo scambio, le scansioni continuano proseguendo gli scambi di posizione degli elementi sinchè i due puntatori non si incontrano nel centro della matrice. A questo punto la suddivisione in due bracci (destro e sinistro) è completa e tutti gli elementi a sinistra sono minori di A mentre quelli a destra sono maggiori.

La matrice, a questo punto delle operazioni, non è ancora a posto, ma è certamente più ordinata e richiederà minor tempo per il successivo trattamento, perchè nessun elemento dovrà essere scambiato tra le due metà (le operazioni di scambio sono quelle che più incidono sul tempo di esecuzione del programma, dunque meno sono meglio è); possono quindi essere riordinate anche separatamente.

Ciò si traduce nell'aver trasformato il problema di ordinare "n" elementi in quello di riordinarne "n/2". Se si prosegue, ripetendo nuovamente il processo di suddivisione, alla fine si ottiene un singolo braccio perfettamente a posto.

La routine di queste pagine usa il metodo descritto con un ulteriore perfezionamento: il "braccio" di matrice selezionato per ogni successiva suddivisione è sempre quello più piccolo e dunque più rapido da gestire.

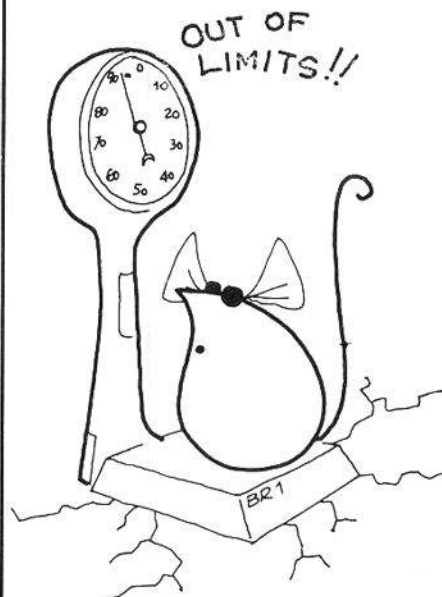
Riportiamo, a titolo di pura curiosità, i dati relativi al confronto dei tempi richiesti dal riordino con quattro algoritmi diversi, per 256 e 512 elementi (rispettivamente, primo e secondo valore tra parentesi):

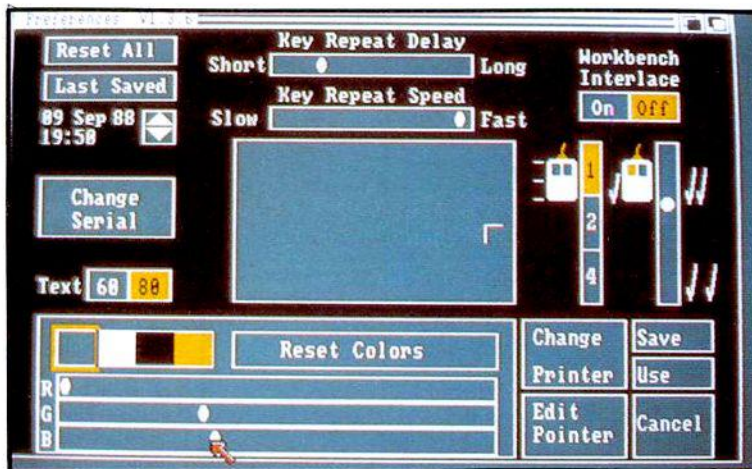
Sort Inserzione (366 / 1444)
Sort Selettivo (509 / 1956)
Sort a Bolle (1026 / 4054)
Quicksort (60 / 146)

La nostra subroutine (o funzione che dir si voglia), per libreria AmigaBasic prevede la chiamata passando semplicemente il nome dell'array interessato ed il numero di elementi da riordinare. Viene invece rilevata automaticamente la dimensione inferiore (zero o uno) della matrice usando la funzione LBOUND, mentre si è evitato l'uso di UBOUND per ricavarne il massimo indice, perchè evidentemente in molti programmi una matrice da riordinare può anche non essere piena.

E' molto semplice adattare la funzione affinchè lavori anche con matrici di stringhe, aggiungendo al nome x(...) nel corpo della funzione il simbolo "$" dopo la "x" ed in tutti gli altri punti di scambio.

Si noti che la matrice deve essere sempre unidimensionale.

```
REM ESEMPIO D'USO ROUTINE DI GIUSTIFICA
DATA "Questa routine formatta delle linee"
DATA "alla lunghezza voluta, inserendo"
DATA "spazi tra le parole come un W.P."
DATA "Troppo corto!!"
FOR j%=1 TO 4
  READ a$
  CALL Justf(a$,40)
  PRINT TAB(10);a$
NEXT
END

' ******************************************************
' * Funzione  Justf(x$,l) ** Libreria per AMIGABASIC *
' *           by Luigi R. Callegari per C.C.C.        *
' * La variabile di stringa 'x$' viene allungata a    *
' * lunghezza 'l' inserendo spazi tra le parole. Non *
' * viene alterata se e  gia  piu  lunga di 'l' e se *
' * e' piu' corta della meta  del valore di 'l'.      *
' ******************************************************

SUB Justf(txt$,l%) STATIC
IF LEN(txt$)>=l% OR LEN(txt$)<(l%/2) THEN EXIT SUB
WHILE  l%<>LEN(txt$)
  FOR j%=1 TO LEN (txt$)
    IF MID$(txt$,j%,1)=" " THEN
      txt$=LEFT$(txt$,j%)+" "+RIGHT$(txt$,LEN(txt$)-j%)
      LET j%=j%+1
      IF l% = LEN(txt$) THEN EXIT SUB
    END IF
  NEXT j%
WEND
END SUB
```

```
REM DIMOSTRATIVO funzione di libreria PINPUT

x$="Prova a non immettere numeri "
y$="0123456789":CALL Pinput(x$,y$,1)
y$="abcdefghijklmnopqrstuvwxyz"
x$= "Non devi immettere minuscole "
Pinput x$,y$,0
END

' ****************************************
' Funzione Pinput(a$,b$,c) per AMIGABASIC
' by Luigi R. Callegari per C.C.C.
' La stringa a$ viene stampata a video e
' attende una immissione da tastiera da
' assegnare alla stessa variabile. La
' variabile 'b$' specifica i caratteri da
' escludere (c=0) o consentiti (c=1) da
' tastiera, secondo il valore di 'c'.
' ****************************************

SUB Pinput(x$,y$,z%) STATIC
  PRINT x$;:x$=""
  IF z%<>1 THEN z%=0
Pinput11:
  k$=INKEY$
  IF k$="" GOTO Pinput11
  IF k$=CHR$(8) THEN
    IF x$="" THEN k$="" ELSE x$=LEFT$(x$,LEN(x$)-1)
    ELSE
      IF k$ = CHR$(13) THEN PRINT:EXIT SUB
      IF INSTR(y$,k$)<>0 THEN i%=1 ELSE i%=0
      IF i%<>z% THEN BEEP:GOTO Pinput11
      x$ = x$ + k$
    END IF
  PRINT k$;:IF LEN(x$)<255 GOTO Pinput11
  BEEP:PRINT
END SUB
```

```
REM ESEMPIO D'USO DI QUICKSORT
WIDTH 78:RANDOMIZE 1:RESTORE
n=512:DIM a(n)
FOR j=0 TO n:a(j)=INT(RND*100):NEXT
PRINT TIME$:Quicksort a(),n:PRINT TIME$
FOR j=0 TO n:PRINT a(j),:NEXT
END

' ******************************************
' * Funzione Quicksort(a(1),n) * AMIGABASIC
' * Di Luigi Callegari per CCC
' * Riordina la matrice a() unidimensionale
' * fino allo elemento a(n).
' ******************************************
SUB Quicksort(x(1),n) STATIC
m = 50:s = 1:DIM stack(m,2)
stack(1,1)=LBOUND(x):stack(1,2)=n
Qsort13:
l = stack(s,1):r = stack(s,2):s = s - 1
Qsort12:
i = l:j = r:a = x(INT((l + r) / 2))
Qsort11:
WHILE x(i) < a: i = i + 1: WEND
WHILE x(j) > a: j = j - 1: WEND
IF i <= j THEN
  SWAP x(i),x(j):i = i + 1:j = j - 1
  IF i <= j GOTO Qsort11
END IF
IF j-l<r-i AND i<r THEN
  s = s + 1:stack(s,1) = i:stack(s,2) = r
  ELSEIF l < j THEN
  s = s + 1:stack(s,1) = l:stack(s,2) = j
END IF
IF j-l < r-i THEN r = j ELSE l = i
IF l < r GOTO Qsort12
IF s <> 0 GOTO Qsort13
END SUB
```

AMIGA CRESCE

Un aggiornamento dei comandi CLI disponibili con la nuova versione del sistema operativo di Amiga

di **Luigi Callegari**

Queste note sono state scritte basandosi sulla versione 1.3 Gamma 4 del Workbench. La Commodore, mentre scriviamo, non ha ancora rilasciato (nemmeno negli USA) la versione definitiva di tale nuovo sistema operativo, perciò esso sarà importato in Italia chissà quando.

Comunque, poichè nel mercato "parallelo" tutti sembrano disporre di tale versione (insieme agli altri tre dischetti del Kickstart per gli A/1000, degli Extras e della documentazione), abbiamo deciso di parlarne, sfruttando la consulenza di insigni programmatori californiani.

I NUOVI COMANDI

Alcuni comandi sono nuovi, altri hanno una nuova sintassi:

COPY

E' forse il comando più potenziato, ma è dotato di un nuovo formato:

COPY [FROM] nome [TO] nome [ALL] [QUIET] [BUF=xxxx] [CLONE] [DATE] [PRO] [COM]

Il parametro BUF = numero permette di specificare quanti buffer di 512 byte ciascuno devono essere usati dal comando (per velocizzare). CLONE è la combinazione di DATE, PRO e COM. DATE copia anche la data del file originale. PRO copia anche i bit di protezione, COM copia anche il commento. Si noti, che la versione 1.2 del comando non eseguiva un CLONE!

DATE

La specifica della data non richiede più uno zero nel caso in cui il giorno sia inferiore a dieci.

DELETE

Quando si specifica con una linea sola la soppressione di più file contemporaneamente, l'esecuzione del comando non si interrompe più quando uno di essi non è stato trovato.

DIR

Presenta una sintassi leggermente modificata:

DIR [nome] [OPT A-I-D] [ALL] [INTER] [DIRS]

La specifica di opzione D sta per DIRS, mentre A ed I (All / Interactive) funzionano come al solito. L'opzione DIRS mostra solo le directory. In modo interattivo (specificando solo il nome del comando), vi è una opzione COMMAND= che permette l'esecuzione di numerosi comandi.

GIA' FATTO

Commodore Computer Club ha già pubblicato una serie di articoli sui comandi del CLI, l'interfaccia convenzionale "a tastiera" di Amiga, collega dell'interfaccia WIMP (Windows, Icons, Mouse and Pointers) chiamata Workbench.

Non possiamo fare a meno di sottolineare che se quest'ultima è più comoda e semplice, il CLI permette un accesso più profondo alla struttura della macchina. La sua conoscenza è alla base di uno sfruttamento completo e redditizio di Amiga.

La serie di articoli sui comandi CLI è stata pubblicata nei numeri di C.C.C. tra il 41 ed il 47 (compresi).

DI VERSIONE IN VERSIONE

Il Kickstart è la parte del sistema operativo di Amiga che veniva caricata da disco (nel vecchio Amiga 1000) ed è invece residente su ROM nei modelli più recenti A/500 e A/2000. La primitiva versione 1.0 di Kickstart non è mai giunta in Italia, per fortuna, in quanto faceva a malapena funzionare il computer.
La versione 1.1 era la versione dei primi A/1000 giunti in Italia. Aveva moltissimi difetti e bugs, ma la documentazione ufficiale della Commodore / Bantam è stata scritta su di essa.
La versione 1.2, attualmente memorizzata nelle ROM degli A500 / A2000, presenta ancora qualche difettuccio, ma certamente è ad un livello accettabile.

La versione 1.3 è quella di cui si parla in queste pagine, ancora in versione pre - release (gamma - version); presenta alcune modifiche non di gran conto, specie per gli utenti dilettanti.
La prossima versione (1.4 o, addirittura, la 2.0) probabilmente uscirà con la nuova generazione di Amiga, di cui si spettegola da tempo, che assomiglierà molto di più a quella del meraviglioso Apple MacIntosh, ma sarà posta in vendita ad una frazione del suo prezzo.
Grazie a nuovi chip custom di Jay Miner (per cui sono già configurati anche gli A500 ed A2000), si potranno, ad esempio, avere 4096 colori contemporaneamente sul video senza limitazioni dello HAM ed altissime risoluzioni senza interlacciamento e senza sfarfalamenti anche su monitor a bassa persistenza. Quasi sicuramente verranno usati i processori 68010 (già previsto dal Kickstart) e/o 68020. Inoltre, si avrà un chip speciale per il controllo hardware del DMA, consentendo di implementare effettivamente su Amiga il sistema operativo UNIX, finora riservato solo a minicomputer da varie decine di milioni di lire.
Si badi che esite un Kickstart V1.3 ed un Workbench V1.3.
Il primo sarà fornito su ROM (vedremo al momento opportuno come installarlo) per gli A500 / A2000 oppure su dischetto per i vecchi A1000. Il secondo, che funziona anche sotto Kickstart 1.2, sarà fornito su dischetto.
Le alterazioni nel Kickstart 1.3 rispetto alla versione 1.2 sono talmente piccole e specifiche (autoboot da hard-disk o periferica) che chi non pensa di comprare un disco fisso per Amiga potrà tranquillamente fare a meno di procurarselo. Tra l'altro, abbiamo già notato che alcuni giochi, scritti evidentemente in maniere non molto convenzionali, non funzionano con il nuovo kickstart V1.3.

DISKDOCTOR
Ora viene verificato se esiste sufficiente memoria libera per il lavoro di Diskdoctor (prima il computer andava in tilt senza dire nulla). L'algoritmo di recupero dei file rovinati è stato migliorato.

ECHO
E' possibile specificare dopo la stringa, nel file batch, l'opzione NOLINE che annulla l'automatico "a capo" alla fine dell'esecuzione del comando.

EXECUTE
Ora questo comando verifica l'esistenza del nome logico T: e lo usa se è assegnato. Altrimenti, viene assunta, per default, la directory ":T".

FF
Questo nuovo comando, prodotto da Charlie Heath della Microsmiths, accelera la visualizzazione dei testi alfanumerici sullo schermo. L'opzione -o (default) attiva questo acceleratore, mentre "-n" lo disattiva.

INFO
Ora si può specificare, dopo INFO, il nome del device di cui si vuole avere informazioni. Ad esempio, INFO DF1:

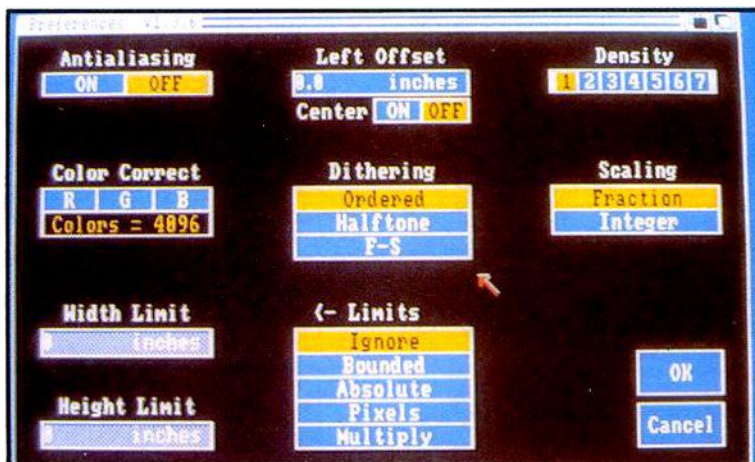
INSTALL
Presenta un nuovo formato e serve anche per proteggersi meglio dai virus:

INSTALL DRIVE DFn: [NOBOOT] [CHECK]

La memoria usata per creare il bootblock su disco è ora azzerata (prima si trovavano nel bootblock anche i contenuti delle ultime linee battute!). Specificando NOBOOT, il disco trattato viene reso non bootable (ovvero, non inseribile al posto del Workbench dopo un reset o all'accensione). Specificando CHECK viene verificato il bootblock del dischetto nel drive indicato, poi viene segnalato se tale bootblock è standard oppure se si sospetta la presenza di un virus.

LIST
E' un comando già noto, ma notevolmente modificato e potenziato. Il nuovo formato è:

LIST [dir] P/PAT [maschera] [KEYS] [NODATES] [QUICK] [BLOCK] [S sottostringa] [SINCE data]] [UPTO data] [TO nome]

Ora vengono mostrati i nuovi quattro bit associati ad ogni file / directory: H (Hidden), S (Script), P (Pure), A (Archive). La loro trattazione, insieme alle nuove caratteristiche dei file sotto V1.3 ed il FFS (Fast File System) saranno trattati in altra sede.

LOCK

Nuovo comando che permette di attivare la protezione di un hard-disk FFS o di una partizione FFS. Permette di usare una parola d'ordine di quattro lettere per chiudere lo hard-disk, od una sua parte, sino allo spegnimento o sino alla riapertura con lo stesso comando e la stessa parola. La sintassi è:

LOCK drive: ON/OFF password

dove "drive" indica il drive interessato, "ON" e "OFF" attivano / disattivano la protezione e "password" è la parola d'ordine.

NEWCLI

La nuova finestra diverrà automaticamente uno shell (nuovo CLI potenziato, di cui riparleremo altrove) se è stato reso residente uno SHELL-SEG. Il formato completo è:

NEWCLI [finestra] [FROM nome]

Se non si specifica il nome del file FROM di inizializzazione della finestra, viene letto per default il file "S:CLI-StartUp", se esistente.

PROTECT

Ora questo comando può aggiungere o togliere (ADD / SUB) bit di protezione da un file. I nuovi bit sono chiamati H, S, P ed A per Hidden, Script, Pure ed Archive.

RESIDENT

Questo prezioso nuovo comando permette di rendere residenti in memoria alcuni comandi (non tutti) del DOS, cosa particolarmente preziosa se si dispone di un solo drive. Il formato è:

RESIDENT [nome] [file] [Delete] [Add] [Replace] [Pure]

SEARCH

Restituisce un codice 0, se l'oggetto è stato trovato, ed un 5 (warn) in caso contrario. Premendo [CTRL]-C si interrompe la ricerca. Il formato nuovo è:

Digitandolo senza parametri, si ottiene l'elenco dei comandi residenti attualmente. Il comando può essere usato solo con quei comandi che sono rientranti e rieseguibili e che abbiano il bit Pure attivato. Inoltre, funziona solo col nuovo Shell della versione 1.3 e non col classico CLI.

RUN

Ora è possibile lanciare un task senza prevenire la chiusura della finestra da cui era stato fatto partire. Ciò viene ottenuto redirigendo ()7 l'output di RUN. Se lo si usa per lanciare un nuovo CLI, viene utilizzato, se presente, il file "S:Cli-Startup" per inizializzarlo.

SEARCH FROM nome SEARCH stringa [ALL] [NONUM] [QUICK]

Specificando l'opzione NONUM, non vengono visualizzati i numeri di linea del luogo in cui è stata trovata la stringa. L'opzione QUICK usa un formato di visualizzazione più rapido e compatto.

SETDATE

Ora, come input di Setdate, si può anche usare direttamente l'output di DATE.

STATUS

Ora si prevedono anche livelli di priorità negativi.

TYPE

Ora viene dato un avvertimento (warning) se il file di destinazione esiste. Il nuovo formato è:

TYPE nome TO nome [OPT H / N / HEX / NUMBER]

Si noti la presenza delle due nuove opzioni HEX e NUMBER.

VERSION

Restituisce il numero di versione di una libreria, di un device o del disco di workbench. Il formato è:

VERSION nome [#VERSIone] [#revisione]

Se non si usano parametri viene semplicemente visualizzato il numero di versione di KickStart e Workbench usati. Specificando un numero minimo di versione e di revisione, esso può attivare dei flag per controllare l'esecuzione di un file batch.

XICON

Si tratta di un nuovo comando, sviluppato nel dominio pubblico americano, che permette di fare eseguire un file batch clickando su di una icona.

PAGE SETTER, IL TIPOGRAFO DI AMIGA

Come trasformare l'Amiga in una piccola tipografia per realizzare, in casa vostra, documenti di ogni tipo

di **Roberto Ferro**

Una volta caricato, il programma mostra sulla destra un ampio Tool Box (letteralmente, "la cassetta degli attrezzi") e 6 menu a diposizione. Il programma è suddiviso in tre sezioni: trattamento testi, trattamento immagini, composizione; fa ampio uso di Intuition e quasi ogni volta che selezioniamo un'opzione viene proposta una finestra di requester tramite cui decidere le modalità di esecuzione della operazione avviata.

Tutto è orientato alla massima intuitività, spesso possiamo anche far a meno del manuale, grazie alla chiarezza delle icone presenti, facilmente interpretabili anche dai neofiti. I passi che conducono alla creazione di un documento sono i seguenti:

- Creazione della pagina/e dalle dimensioni desiderate;
- Immissione del testo da trattare mediante l'apposito Text Editor o, in alternativa, importandolo da un altro Word Processor (esempio, Scribble!);
- Immissione delle figure da creare con il Graphic Editor (incluso), oppure importandole da altro programma grafico (Page Setter supporta il formato IFF e sostituisce i colori con retini bianco / nero);
- Disposizione del testo e delle immagini sulla pagina, a seconda delle esigenze;
- Ripetizione della procedura per il numero di pagine stabilite;
- Stampa!

Tutto il processo richiede una certa dimestichezza con l'uso del programma e presuppone una certa abilità nella composizione; dopo qualche tentativo tutto diventa piuttosto semplice.

LA PRIMA PAGINA

Per creare una pagina dovremo scegliere, dal menu Page, l'opzione Create. Verrà su-

AMIGA: IMPAGINAZIONE ELETTRONICA

Un esempio di utilizzo di Page Setter, l'ottimo programma di DeskTop Publishing per Amiga

di *Roberto Ferro*

La presente pagina e' stata composta con Page Setter, ed e' solo un semplice esempio di come si puo' presentare un documento realizzato con questo package. Le immagini fanno parte della libreria di disegni disponibili sul disco nella directory ClipArt.
La presente pagina e' stata composta con Page Setter, ed e' solo un semplice esempio di come si puo' presentare un documento realizzato con questo package. Le

immagini fanno parte della libreria di disegni disponibili sul disco nella directory ClipArt.
La presente pagina e' stata composta con Page Setter, ed e' solo un semplice esempio di come si puo' presentare un documento realizzato con questo package. Le immagini fanno parte

della libreria di disegni disponibili sul disco nella directory ClipArt.
La presente pagina e' stata composta con Page Setter, ed e' solo un semplice esempio di come si puo' presentare un documento realizzato con questo package. Le immagini fanno parte della libreria di

In queste pagine: alcuni esempi di impaginazione elettronica realizzabili con Page Setter per Amiga. Facile è l'impostazione del numero di colonne, delle immagini e del testo; altrettanto semplice risulta lo spostamento di varie parti del documento.

bito proposto un requester per decidere le dimensioni della pagina (A4, A6, Standard, Legal o qualunque altra dimensione). Possiamo anche decidere di creare più pagine, tutte con le stesse caratteristiche. In questo caso faremo avanzare il contatore di pagine, contenuto nello stesso requester, fino al numero desiderato. Si termina, ovviamente, clickando su OK.

La pagina, vuota, verrà quindi mostrata interamente sullo schermo. Attraverso il gadget a forma di lente possiamo scegliere tre tipi di ingrandimento, mentre attraverso il gadget proporzionale (posto sotto quest'ultimo) spostiamo lo sguardo sulla zona della pagina che più interessa.

Per ciò che riguarda le facilitazioni, attraverso l'opzione Grid Spacing del menu Art Board possiamo decidere la grandezza della griglia da impostare. Questa può variare da 1/8 a 1/2 di pollice ed aiuta nell'individuare varie zone della pagina. Allo stesso modo possiamo decidere la gradazione dei righelli, da porre in alto e a lato della pagina stessa.

Sia la griglia, sia i righelli, sono disinseribili con apposite opzioni del tool box, ma è anche possibile, attraverso l'icona con la freccetta, disattivare la griglia pur mantenendola visibile; saremo quindi liberi di posizionare testi e immagini senza seguire le spaziature imposte da quest'ultima.

Dal menu Page possiamo anche scegliere l'opzione Default mediante la quale impostare i parametri standard per la creazione di una nuova pagina, opzione utilissima quando dobbiamo creare più pagine con parametri particolari ma con lievi differenze. Creata la pagina, si può dare inizio alla "composizione", ma è bene, dapprima, chiarire il concetto di box (o casella).

I BOX

Page Setter permette di creare tanti box all'interno di una pagina, ognuno dei quali idoneo a contenere testi oppure disegni (ma non entrambi). Una volta riempiti i box con ciò che vogliamo, possiamo accostarli o allontanarli tra loro come più ci aggrada: un ottimo sistema per unire testo e grafica in una pagina senza combinare pasticci.

Per creare un box selezioniamo il gadget a forma di cubo presente nel tool box e ci portiamo sulla pagina dove ne stabiliamo le dimensioni. Il box verrà sempre creato con caratteristiche standard. Per modificarle è necessario agire come per la pagina: selezionando Default dal menu Box. Page Setter offre un ampio set di possibilità, tra cui la scelta del font di caratteri da utilizzare, oppure la regolazione della giustificazione del testo (che può essere allineato a destra, a sinistra, entrambi o centrata). Si può scegliere la distanza dal margine superiore e sinistro del testo, nonchè l'interlinea. Possiamo ancora decidere se visulizzare un box con bordo (oppure no), lo spessore dello stesso, oppure impostare un retino per lo sfondo.

Un box può essere modificato, in grandezza, anche dopo la sua creazione clickando nell'angolo in basso a destra. Possiamo anche spostarlo, clickando in qualunque altra parte.

IMMISSIONE TESTI

Per l'elaborazione dei testi bisogna spostarsi, nella sezione di programma dedicata, mediante l'opzione Text Editor del menu Department. Ci si ritrova di fronte ad un "ambiente" che ricorda molto "Scribble!".

se non altro per il suo aspetto molto spartano.

Se esiste già, su disco, un file digitato con un altro word processor, non resta che caricarlo: Page Setter riconosce i file Scribble! e TextCraft di cui conserva i codici di formattazione.

Sul Text Editor non ci soffermiamo a lungo; si tratta di un programma che non fa miracoli, ma ha tutto l'indipensabile per far bella figura e risulta piuttosto rapido negli scorrimenti del testo. Vi sono codici per identificare lo stile dei caratteri: "/b" per il boldface, "/u" per il sottolineato, "/s" per l'effetto ombra (molto carino), "/i" per l'italico e "/o" per outlined. E' possibile anche combinare tra loro i vari codici ottenendo una ampia varietà di stili.

Con Exit ci si ritrova nel quadro pincipale. Per visualizzare il testo in una casella si seleziona il gadget con la penna e si clicka in un box. Se il testo non può essere contenuto per intero in una sola casella, possiamo farlo continuare in un'altra attravero l'apposita opzione di concatenamento del ToolBox che ha il compito di collegare, appunto, due o più caselle tra loro.

Il font utilizzato per il testo sarà quello attivo per default al momento della creazione del box, ma possiamo comunque modificarlo, insieme agli altri parametri, con l'opzione Current del menu Box. Perfino dopo l'immissione del testo si può modificare la grandezza del box: il testo vi si adatterà automaticamente.

A PROPOSITO DEI "SUPER" WORD PROCESSOR

Oramai la febbre del DeskTop Publishing (DTP per gli amici) ha coinvolto tutti. Dallo studente al professionista, dalla segretaria all'impiegato, tutti trovano il modo di sfruttare al meglio, e secondo le proprie necessità, il nuovo campo di applicazione del personal computing.

Per dare un'idea delle dimensioni del fenomeno, basta ricordare che nel mondo sono stati venduti circa 300.000 pacchetti per il DTP nel 1987, con un volume di affari che sfiora i 500 milioni di dollari (hardware compreso), pirateria a parte; oppure basta pensare che un computer come il Macintosh deve la sua fortuna proprio alla introduzione di software per il DTP ad esso dedicati.

Non c'è sistema operativo che non vanti il suo package di alto livello per il DTP: PageMaker e ReadySetGo!4 (su Machintosh); ancora Pagemaker e Ventura su MS/DOS, Fleet Street Publisher per Atari ST e, dulcis in fundo, Professional Page e Shakespeare per Amiga.

"Bene!" dirà qulcuno "vediamo cosa possono fare questi programmi." Invece no; ci occuperemo, questo mese, di "Page Setter" della Gold Disk (la stessa che produce il fratello maggiore "Professional Page") perchè, trovandosi sul mercato da parecchio tempo, è uno dei podotti più affidabili in commercio, presenta una vastità di opzioni più che sufficiente per un DTP "casalingo" (quello di studenti e professionisti per intenderci), non ci costringe all'acquisto di una costosissima stampante laser per ottenere un documento accettabile, è reperibile assai facilmente, in inglese ma anche in italiano ed esiste la versione PAL che sfrutta le 256 linee di risoluzione dei monitor europei contro le 200 americane.

Chiariamo subito che, desiderando un pacchetto per esigenze spiccatamente professionali, è bene rivolgersi ad altri prodotti più potenti, meglio documentati e ampiamente supportati dalla casa di produzione. Tra questi suggeriamo i già citati Professional Page (output su stampante Laser a 300 X 300 Dot Per Inch) e Shakespeare (output su Laser ma anche su una coloratissima InkJet) entrambi blasonati ed affidabili; il primo è della Gold Disk e il secondo della Infinity Software.

IMMISSIONE GRAFICI

La procedura per porre disegni sulla pagina è quasi identica a quella appena descritta per il testo. Selezionando Graphic Editor dal menu Department entriamo in un editor piuttosto povero di opzioni, ma, come già detto, è possibile importare immagini IFF (incluso HAM) che verranno rese con diversi retini. Creata l'immagine è necessario clip-arla selezionando il gadget con il quadrato e riquadrando la parte che interessa; in seguito si ricorre all'opzione Clip.

Possiamo quindi scegliere la scala 1:1 oppure 5:4 la quale, modificando le proporzioni del disegno, riduce l'effetto di deformazione, che normalmente otteniamo in fase di stampa.

"Uscendo" con Exit si usa il pennello per collocare l'immagine realizzata all'interno del box desiderato, adoperando le quattro freccette (poste in basso nel tool box) per posizionare correttamente l'immagine all'inteno del box.

PREFERENCES

Tra i menu a disposizione ve ne è uno, Preferences, che permette di modificre le condizioni di lavoro: possiamo alterare i colori del testo, dello sfondo e della griglia, attivare un Memory Warning che avvertirà nel caso in cui dovessimo trovarci in ristrettezze... mnemoniche, oppure selezionare il Memory Cleanup che risistemerà tutti i dati in modo da liberare spazi occupati inutilmente (funzione che dovrebbe essere presente in ogni programma a causa del pessimo meccanismo di Cleanup provvisto dall'AmigaOS). Una funzione molto utile è anche Quick Move che consente un rapido spostamento dei box anche quando sono grandi e "pieni" di testo e grafica.

LA STAMPA

Selezionando l'opzione Press (dal menu Department) diamo il via alla fase di stampa che terrà conto del driver (= programma specifico per pilotare la nostra stampante) e della stampante installata.

Si possono istallare molti tipi di stampanti, purchè a matrice di punti e supportate da un driver già presente nelle Preferences. A questo proposito segnaliamo che, con il nuovo Workbench v1.3, sono stati introdotti driver migliorati nella resa e nella velocità anche per stampanti a 24 aghi. Il trio Page Setter + Driver per 24 aghi (EpsonQ nella fattispecie) + Stampante a 24 aghi con risoluzione di 360 X 360 DPI (maggiore, cioè, delle laser) conferisce una resa veramente professionale.

E' chiaro che non è tutto oro quel che luccica; non tutte le 24 aghi producono ottimi output. In linea di massima sembrano di ottima qualità gli output delle stampanti Nec serie PinWriter o Citizen serie HQP (ma ve ne sono molte altre).

Un secondo modo per effettuare la stampa è l'utilizzo dell'utility PagePrint. In questo caso bisogna salvare il documento; in seguito si passa al Workbench. Da qui si clicka una volta sull'icona del documento e poi, tenendo premuto il taso shift, lanciando PagePrint il quale aprirà un task che, agendo in background (senza cioè mostrarsi), stamperà il file senza tenere occupato l'Amiga. Magie del multitasking...

Infine c'è un metodo più complesso, che val la pena esaminare. Con un'altra utility presente sul disco del Page Setter (ci riferiamo a PageIFF), possiamo trasformare

Stili

Testo normale

Testo in Boldface

Testo Sottolineato

Testo in Outline

Testo Ombreggiato

Testo in combinazione

Ombreggiato e Outline

Boldface sottolineato

Outline in Boldface

Boldface sdoppiato

Tutti gli stili di Page Setter. Notare la varietà disponibile e la possibilità di combinare più opzioni tra loro.

Desktop Publishing per *AMIGA*

PageSetter e Amiga: una accoppiata vincente per il DTP personale. Un sistema nuovo pratico ed economico.
PageSetter e Amiga: una accoppiata vincente per il DTP personale. Un sistema nuovo pratico ed economico.
PageSetter e Amiga: una accoppiata vincente per il DTP personale. Un sistema nuovo pratico ed economico.

PageSetter e Amiga: una accoppiata vincente per il DTP personale. Un sistema nuovo pratico ed economico.
PageSetter e Amiga: una accoppiata vincente per

il DTP personale. Un sistema nuovo pratico ed economico.
PageSetter e Amiga: una accoppiata vincente per il DTP personale. Un sistema nuovo pratico ed economico.
PageSetter e Amiga: una

un qualsiasi documento in un file IFF, formato notoriamente caricabile da un qualunque programma grafico (DPaint, Graphicraft, Express Paint, Photon Paint...) e modificabile, magari aggiungendo un po' di colori.

Il risultato ottenuto in tal modo può essere stampato (sempre come schermata IFF) su una qualunque stampante a colori con risultati apprezzabili.

PERTANTO...

...procuratevi Page Setter!

Scherzi a parte, Page Setter è veramente un programma formidabile, soprattutto considerato nell'ottica dell'uso personale. Certo non ha le centinaia di opzioni di programmi come Professional Page o i tanto decantati Ventura e Pagemaker, ma di certo può vantare una semplicità d'uso infinitamente superiore e non ha bisogno di hard disk (il programma è piccolino; due drive sono più che sufficienti). Se, poi, ci pensate bene, vedrete che non è indispensabile avere righelli mobili, fantastici ingrandimenti oppure disporre di mirabolanti operazioni di scorrimento del testo attorno ad una immagine.

Page Setter è, in definitiva, l'ideale per tutti coloro che non hanno eccessive pretese, non ritengono indispensabile la qualità della laser e si accontentano dell'uso di una stampante a matrice di punti, risparmiando (molto) in denaro e guadagnando altrettanto in praticità d'uso.

L'AMIGA INFEDELE

Raccolta di alcuni errori riscontrati nel linguaggio interprete AmigaBasic; ed un paio di programmini di utilità generale

di **Luigi Callegari**

MEMORIA

Quando si carica AmigaBasic, automaticamente il sistema operativo, in un modello standard, azzera 25K di Ram e li riserva per suo uso e consumo. Questo evento viene segnalato dal messaggio iniziale che appare nella finestra dei comandi (sotto il copyright).

In realtà è disponibile una maggior quantità di memoria, ovvero (quasi) tutta quella non riservata al sistema ed all'interprete. Il comando per richiedere tale memoria "aggiuntiva" per il nostro programma è:

```
CLEAR, x
```

Dove la virgola è obbligatoria e il valore X indica il numero di byte di cui vogliamo disporre (ad esempio: Clear, 100000). Questo è quanto dice il manuale, ma è bene usare un doppio comando sulla stessa linea, specificando prima la quantità standard e poi quella necessaria. Per disporre di 100 K, ad esempio,...

```
Clear, 25000: Clear, 100000
```

Infatti se si usa solo il comando "Clear, 100000" all'interno di un programma, questo si bloccherà (o andrà in crash) quando si tenta per la seconda volta l'esecuzione dello stesso, in quanto si richiede l'azzeramento di una grossa quantità di memoria per due volte. Amigabasic, infatti, non riparte ad ogni nuovo RUN con la quantità di memoria di default (25K) perchè, al termine dell'esecuzione di un programma, non rilascia al sistema la memoria allocata dall'ultimo Clear.

Quando nella documentazione di un programma scritto in AmigaBasic si trova che è bene dare un "Clear, xxxxx" in modo diretto prima dell'esecuzione, è possibile provare ad inserire, come prima linea, il doppio Clear come spiegato.

STRINGHE LUNGHE

Una variabile di stringa (ovvero, alfanumerica) in AmigaBasic può avere sino a 32767 caratteri, in contrasto con i tipici basic delle macchine ad otto bit (C/64) che potevano averne solo 255.

Comunque non è effettivamente possibile disporre di una stringa così lunga se prima non si è azzerata (con un Clear) una certa quantità di memoria idonea a contenere la stringa stessa ed il programma.

In questo modo diventa possibile realizzare archivi in cui un lungo file è contenuto in una sola variabile. Ad esempio:

```
Clear ,100000
While 1
Input "Nome "; A$
If A$ = "" goto Ricerca
Input "Telefono: "; b$
File$ = file$ + a$ + b$
Wend
Ricerca:
While 1
Input "Nominativo da cercare "; A$
If A$ = "" then end
N = instr (File$, a$)
If N=0 then print "Non c'è!" Else Print
Mid$ (file$, n, 30)
Wend
```

Questo programma mostra, appunto, come sia possibile creare un intero archivio in una sola variabile, sfruttando poi la funzione INSTR per fare eseguire al Basic direttamente le ricerche. E' molto rudimentale, ma serve al suo scopo e vale anche come esempio illustrativo.

OROLOGIO

Sfruttando le istruzioni di gestione dell'ora (DATE$ e TIME$), quelle di gestione temporizzata (ON TIMER GOSUB, TIMER) e quelle di gestione delle finestre (WINDOW) è possibile, con poche linee, avere un orologio sempre visibile nei nostri programmi Basic, concorrentemente all'esecuzione degli stessi. Vediamo come:

```
rem Esempio di orologio
window 12, "clock", (10, 10) - (50, 20),
18
window output 1
on timer (29) gosub PrintClock
Timer on
rem .....Programma principale ....
End

PrintClock:
attw = window(1)
window output 12
locate 1, 1
print left$ (time$, 6);
window output (attw)
Return
```

Si noti che la finestra di orologio può essere spostata (ma non ridimensionata) e che si è tenuto conto dell'eventuale uso da parte del programma di altre finestre di output, a parte la numero uno, di default, selezionata inizialmente.

La funzione WINDOW(1) restituisce il numero di finestra correntemente attiva, cioè al momento del servizio della richiesta di interruzione (ogni trenta secondi, come specifica il numero nella linea ON TIMER), che viene ri-selezionata, dopo la scrittura in finestra 12 dell'orologio, con WINDOW OUTPUT.

PULCI

Se la versione 1.1 di Amigabasic era praticamente inutilizzabile per i numerosi bug presenti, la versione 1.2 ne ha pochi, ma "buoni".

Avendo programmato a lungo con questo linguaggio abbiamo scoperto che, inspiegabilmente, salvando un programma mediante l'apposita opzione di menu (SAVE oppure SAVE AS) talvolta viene registrata sul disco solo una piccola porzione del testo: quando si ricarica il programma ne risulta quindi persa una considerevole quantità.

Altro bug si verifica usando il TRACE per visualizzare la linea eseguita nel programma. Se si attiva tale opzione e poi il programma si ferma (per un errore od un comando STOP / END) non si deve disattivare il modo TRACE pena un quasi certo crash della macchina.

E' bene inoltre che il modo TRACE sia disabilitato quando si salva il programma su dischetto.

VIRUS, UN DOTTORE PER AMIGA

Abbiamo spesso parlato di virus informatici. Poichè il fenomeno, legato alla circolazione di software copiato illegalmente, sta assumendo dimensioni colossali, è bene spendere qualche parola in più

di **Luigi Callegari**

Il sistema operativo di Amiga costituisce il terreno ideale per la nascita e la proliferazione di virus. Un programma di questo tipo, infatti, può sfruttare le caratteristiche di auto-boot dei dischi di Amiga (come il Workbench) ed il multitasking sia per installarsi sia per moltiplicarsi velocemente.

E tutto questo senza che, ovviamente, sia possibile accorgersi di nulla, perchè la maggior parte dei virus di Amiga (ne conosciamo già una mezza dozzina) non segnalano la propria presenza prima di essere usati un certo numero di volte.

Un virus prodotto dalla SCA si segnala, ad esempio, resettando lo stesso programma (pigiando i tasti CTRL ed i due Amiga contemporaneamente) per sedici volte, non prima.

I virus, anche se invisibili, "vivono" e, una volta caricati in memoria dal dischetto al momento del boot iniziale (quando si inserisce il disco alla richiesta del Workbench) provvedono a duplicarsi su eventuali dischetti presenti nei drive esterni, oppure negli stessi dischi introdotti in quello interno, controllando, ad intervalli di tempo regolari, l'eventuale sostituzione dei dischetti

da parte dell'utente. Quando si accorgono del cambio di un disco, subito i virus vi scrivono una copia di se stessi, sempre dotata di auto-boot.

Questo lavoro, che solitamente impone la scrittura di particolari routine di servizio delle interruzioni negli altri computer, in Amiga sfrutta la possibilità normale di ogni "task" di essere eseguito insieme ad altri. Il nostro virus risulta quindi anche estremamente compatto, occupa poca memoria, e può infettare dischetti anche mentre stiamo lavorando con un word processor o un database, senza che possiamo accorgerci del LED del drive che si accende per un secondo senza apparenti motivi e senza farci sospettare che il virus si è appena clonato...

COME CI SI ACCORGE DEL VIRUS

I virus sono particolarmente nocivi per quei programmi commerciali protetti che mandano in tilt il computer se trovano alterato anche un solo byte sul dischetto (al fine di proteggersi da copie illegali). I virus, infatti, "si" scrivono indiscriminatamente e senza controllo nei primissimi blocchi del disco, alterando comunque il suo contenuto.

Inoltre l'infezione di un dischetto lo rende "bootable", ovvero inseribile alla richiesta del Workbench da parte del sistema. Se, pertanto, abbiamo un dischetto che non era bootable (magari appena formattato) e lo diventa inspiegabilmente, certamente ci troviamo in presenza di un virus, entrato in circolazione "grazie" all'uso precedente di altri dischetti infetti.

Per controllare direttamente se è presente un virus su di un dischetto, si può procedere come segue:

1- Accendere il computer dopo averlo tenuto spento per un paio di minuti.

2- Alla richiesta del Workbench inserire il dischetto da verificare.

3) Permettere che il led del drive si accenda per la durata tipica di un secondo (circa); non appena si spegne, resettare subito l'Amiga con i tre soliti tasti e si tenga pigiato il pulsante sinistro del mouse.

4- Se lo schermo vuoto, durante il reset, vira ad un colore verde scuro (invece di una serie di bianchi - grigi) vuol dire che c'è un virus su quel dischetto, caricato in memoria nell'istante del boot.

Tale sistema funziona per tutti i virus attualmente conosciuti, ma non necessariamente per tutti quelli esistenti o che potrebbero essere scritti in futuro.

La diagnosi più professionale viene comunque eseguita usando un editor di dischetti tipo SMARTDISK (ovvero uno di quei programmi che permettono di leggere direttamente i contenuti del dischetto) ed analizzare i primi due blocchi. Se, al loro interno, si trovano i codici ASCII di messaggi virulenti, il gioco è fatto. Se non si trova nulla, si può stare (sufficientemente) tranquilli, perchè il virus deve risiedere forzatamente in questi blocchi, che sono quelli caricati al momento del boot.

A meno che non si tratti di un virus che non preveda messaggi, ma la cosa è fortunatamente molto difficile, grazie alla megalomania degli "untori".

COME SI GUARISCE DAL VIRUS

Esistono un paio di programmi di pubblico dominio, ovvero divulgati gratuitamente in USA (un po' meno in Italia) che permettono di rilevare e sopprimere con un semplice tocco di tasto il virus dai dischetti (previa autorizzazione alla scrittura).

Nel caso in cui il programma "infettato" avesse un formato "strano" (o protetto) ed il virus abbia ormai corrotto alcuni dati determinanti (insolitamente memorizzati nei primi due blocchi del dischetto), il nostro programma è purtroppo perso in ogni caso perchè il virus killer non può ricostruire ciò che il virus ha distrutto sovrascrivendo.

Oppure, metodo tanto efficace quanto banale, è di usare un comando INSTALL da CLI. Nel caso si possieda solo il drive interno, la sequenza di operazioni da eseguire per decontaminare una serie di dischetti è la seguente:

1- Attivare il CLI selezionandolo, da Workbench, con una doppia pressione del mouse sulla sua icona.
2- Rimuovere l'eventuale protezione da scrittura dai dischetti da decontaminare.
3- Digitare la seguente linea CLI:

COPY C/INSTALL TO RAM:

4- Inserire nel drive interno il dischetto da disinfestare.
5- Digitare la linea CLI:

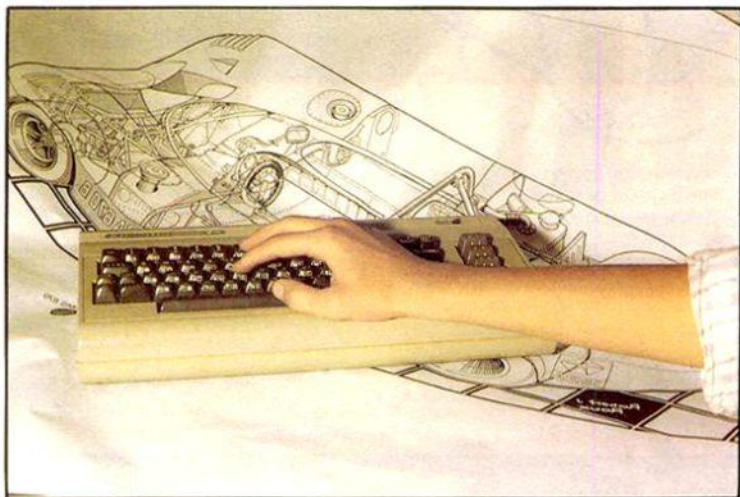
RAM:INSTALL DFO:

Solo gli ultimi due passi (4 e 5) devono essere ripetuti per tutti i dischetti da trattare.

COME EVITARE INFEZIONI DA VIRUS

1- Verificare, con i sistemi prima descritti, lo stato di "salute" del nostro archivio di dischetti. Siate pignoli: basta un solo dischetto inquinato per rovinare in pochi giorni tutti gli altri.

2- Tenere tutti i dischetti importanti, e soprattutto quelli originali, sempre protetti dalla scrittura.

3- Ogni volta che si entra in possesso di nuovi programmi copiati (o di dubbia provenienza), verificare subito l'eventuale presenza di virus e, in caso positivo, provvedere immediatamente al trattamento sterilizzante. Provvedete poi a ripulire anche il disco del Workbench o del DOS, nel caso si sia usata la procedura CLI-INSTALL.

TESTO E GRAFICA FINALMENTE INSIEME

Visualizzare messaggi sulla pagina grafica del C/64 è sempre stato un problema; in queste pagine viene descritto un semplice metodo per risolverlo

di **Giancarlo Mariani**

Il Commodore 64, come altri home computer del genere, gestisce in modo diverso lo schermo a seconda se ci si trova in modo testo oppure in modo grafico.

Nel primo caso, infatti, l'uscita video sarà su 40 colonne (80 nel 128), 25 righe, e l'elemento più piccolo di visualizzazione è il carattere.

In modo grafico, invece, lo schermo viene gestito come una matrice di punti, e precisamente, se consideriamo che un carattere è formato da 8x8 punti e che su una riga vi sono 40 caratteri, avremo 40x8=320 punti orizzontali e 25x8=200 punti verticali, per un totale di 64000 punti, o "pixel" (elemento più piccolo in modo grafico).

Come si può intuire, il modo di gestione dello schermo è notevolmente diverso nei due casi; ne consegue che le routine che si occupano della visualizzazione dei dati devono essere altrettanto diverse. Da questo se ne deduce che, ad esempio, non è possibile impartire un comando del tipo "PRINT" quando si è in modo grafico e, viceversa, in modo testo non è possibile "accendere" punti a piacere sullo schermo.

Più di una volta, tuttavia, si avverte l'esigenza di avere, sullo schermo grafico, qualche messaggio da visualizzare, che descriva il disegno o il grafico raffigurato. Nei Tool del tipo del Simon's Basic (ed anche nel nostro GWbasic Emulator) esistono istruzioni che si comportano più o meno come PRINT, ma che agiscono sulla pagina grafica.

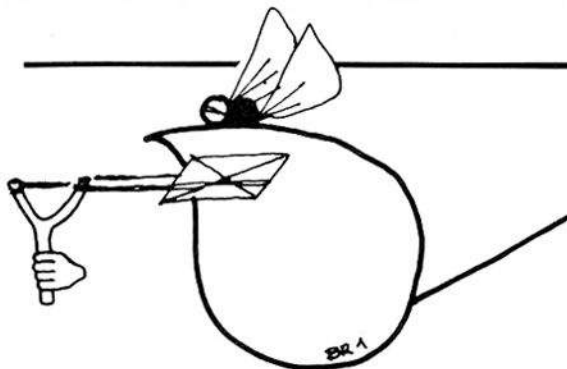
"Più o meno", significa che non tutte le funzioni della normale PRINT sono accessibili (tanto per citarne una, la TAB); sarebbe comodo, quindi, disporre di un sistema per gestire la pagina grafica proprio come se fosse uno schermo in modo testo.

A questo problema si rimedia tramite la routine qui proposta, di cui ora descriviamo le caratteristiche.

Invece di implementare un comando PRINT in grafica, che abbia tutte le caratteristiche del corrispondente comando in modo testo, si risolve il problema direttamente alla radice: si scrive tutto ciò che interessa sullo schermo in modo testo, utilizzando comandi "normali" del Basic (PRINT, POKE, e così via); in seguito, tramite una semplice SYS, si trasferisce l'intero schermo "testo" in quello "grafico", mantenendone inalterato il contenuto.

In pratica la routine non fa altro che leggere, carattere per carattere, il contenuto dello schermo in modo testo; subito dopo aver elaborato opportune conversioni, lo trasferisce nella zona di memoria che contiene la pagina grafica.

COME FUNZIONA

Il funzionamento della routine è semplice: dapprima legge il contenuto del primo carattere della memoria video (ricordiamo

che inizia da $0400), ne individua il codice e ne calcola l'indirizzo del generatore di caratteri; a partire da quest'ultimo legge 8 byte e li trasferisce nei byte della pagina grafica che corrispondono al carattere presente nella pagina testo.

A questo punto non resta altro che incrementare la lettura dello schermo (ossia, leggere il secondo carattere), e ripetere il ciclo descritto fino alla locazione 2023 (ultima locazione della pagina testo).

Caricate il GWbasic e, subito dopo, la routine di queste pagine a partire da un indirizzo libero (esempio: 49152); quindi digitate il seguente programmino:

```
100 cls 0: for i=1 to 100
110 print "a" chr$(32);: next
220 screen 1: cls 1
330 fori =30 to 120 step 30
440 circle (100,100),i: next
550 sys 49152,1
660 goto 660
```

Impartito il Run vedrete un centinaio di "A" sullo schermo in modo testo; subito dopo verranno tracciate quattro circonferenze sullo schermo grafico ed attuato il "trasferimento" delle "A".

La routine ha anche due parametri opzionali che consentono di lavorare con qualsiasi programma grafico ed in qualsiasi condizione. Il primo specifica la sovrapposizione, ossia se il trasferimento dello schermo di testo deve cancellare ciò che c'era prima in pagina grafica oppure no (notare il "virgola" 1 in riga 550). L'effetto è ottenuto leggendo prima il contenuto della pagina grafica e quindi effettuando un OR con il byte del carattere.

DAI BYTE ALLO SCHERMO

Le fasi da seguire per la conversione da testo a grafica non sono complesse, ma richiedono una certa conoscenza del modo in cui è organizzata la memoria della macchina e la gestione degli stessi caratteri.

Un carattere, qualunque esso sia (lettere, numeri, simboli semigrafici) è racchiuso in una griglia di 8x8 puntini. Il C/64, ovviamente, deve "sapere" da dove prelevarli, per poi visualizzare correttamente il carattere nella pagina grafica.

A questo scopo è necessario ricordare che esiste una particolare zona di memoria ROM, lunga 4k, in cui sono memorizzate tutte le informazioni relative a ciascun carattere visualizzabile (maiuscolo, minuscolo, semigrafico, reverse).

Questa zona è così composta: ogni carattere, essendo formato da 64 punti, necessita di 64 "elementi di memoria" (bit) per la sua memorizzazione. Siccome ogni byte di memoria contiene 8 bit, ne consegue che un carattere ha bisogno di 8 byte. Il primo byte (cioè i primi 8 bit), conterrà le informazioni relative alla prima riga del carattere, il secondo alla seconda, e così via, tenendo conto che ogni bit posto al volore unitario, significa punto acceso, ossia visualizzato, mentre ogni bit posto a 0 è un punto spento.

Per fare un esempio pratico, una "E" maiuscola potrebbe essere composta come in figura.

I puntini rappresentano i bit posti a 0, mentre gli "1" a 1. Ovviamente ogni sequenza di 8 bit andrà convertita in decimale, in modo da ottenere il valore del byte corrispondente, seguendo tecniche sulle quali non ci intratteniamo.

In altre parole, nella ROM del generatore di caratteri questi ultimi saranno memorizzati proprio come esaminato, ossia con una sequenza di 8 byte. L'ordine con cui i caratteri sono memorizzati segue quello dei codici della memoria di schermo (descritti sul manuale del computer). Dal momento che al codice 0 della memoria di schermo corrisponde la chiocciolina (tasto presente tra la freccia in alto e l'asterisco), i primi 8 byte della ROM del generatore di caratteri saranno dedicati proprio a tale simbolo. I successivi 8 byte comporranno la "A" (codice di schermo = 1), quindi i successivi 8 la "B", poi la "C", e così via per tutti i 256 caratteri.

L'operazione da compiere, per rintracciare la composizione di un carattere nel generatore, è semplicissima: chiamando "G" l'indirizzo di partenza della ROM del generatore, segue che...

Ind. Car. = G + 8 * (Cod. schermo carattere)

Abbiamo precedentemente affermato che la ROM del generatore è "lunga" 4k; un rapido calcolo, invece, indica che lo spazio necessario per 255 caratteri è 255x8 byte=2048 byte, ossia 2k.

Sembrerebbe che 2k di ROM vadano sprecati, ma ciò non è vero dal momento che vengono utilizzati per decodificare il set minuscolo / maiuscolo; ricordiamo, infatti, che il C/64 possiede, al suo interno, due distinti set di caratteri, intercambiabili tramite i tasti shift e Commodore premuti contemporaneamente. Il primo è il set maiuscolo / grafico presente all'accensione; l'altro è quello minuscolo / maiuscolo.

La memoria in cui è presente la Rom di cui stiamo parlando è allocata, per 4k, a partire dall'indirizzo esadecimale $D000 (decimale 53248).

Il lettore esperto sa benissimo, tuttavia, che in questa zona sono già contenuti gli indirizzi e le porte di I/O del computer, quali le coordinate di sprite, suono, colori, e così via. Il "mistero" della doppia "capienza" consiste nel fatto che il generatore di caratteri è una delle ROM "nascoste" del C/64, che c'è ma non si vede: per accedervi è necessario modificare la configurazione del processore 6510 tramite manipolazioni della locazione 1.

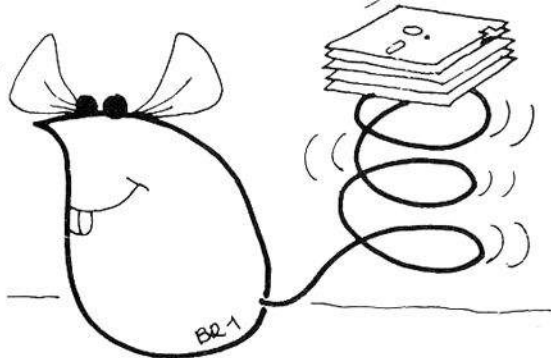
In pratica, per consentire al processore di "vedere" questa ROM, bisogna azzerare il bit 3 della locazione 1, tramite le operazioni l.m.:

Il secondo parametro opzionale specifica l'indirizzo di partenza della pagina grafica che, per default, è posto a $E000 (57344).

La sintassi completa della routine è quindi la seguente:

SYS XXXX, S, PAG

in cui:
XXXX: è l'ndirizzo di partenza della routine stessa.
S: indica la sovrapposizione (0 = Cancella, 1 = Sovrappone. Default=0)
PAG: indica l'indirizzo della pagina grafica (Default = $E000)

Le ultime note riguardano i limiti della routine: essa infatti non può trasferire il testo su di una pagina grafica in multicolor, ma solo in schermate ad alta risoluzione (320x200 punti). Inoltre è necessario prestare la massima attenzione nell'indicare l'indirizzo della pagina grafica; se, per errore, trasferite in pagina 0 oppure "sul" programma Basic, o in altre locazioni pericolose, il computer si inchioderà irrimediabilmente.

LA SINTASSI

Dopo aver caricato la routine LM proposta, per farla funzionare ogni volta che lo desideriamo è sufficiente impartire SYS XXXX, in cui XXXX è l'indirizzo di partenza assegnato (ricordiamo che la routine, come TUTTE quelle dell'enciclopedia, è interamente rilocabile).

La routine è stata prevista per schermate grafiche di programmi come Simon's Basic o GWbasic, che hanno la pagina grafica a partire dall'indirizzo $E000; tuttavia, come vedremo, sarà possibile modificare questo indirizzo per adattare la routine a qualsiasi altro programma grafico.

```
LDA $01
AND #$FB
STA $01
```

Oppure, meno prudentemente, in Basic:

```
Poke 1, Peek(1) and 252
```

Non bisogna dimenticare comunque di disabilitare l'interrupt prima di svolgere questa operazione, tramite l'istruzione l.m.:

```
SEI
```

...ed in Basic:

```
Poke 56334, Peek(56334) and 254
```

A questo punto si può leggere il contenuto del generatore di caratteri, per poi riabilitare, alla fine delle operazioni, sia gli indirizzi di I/O che l'interrupt, tramite le istruzioni:

```
LDA #$01
ORA #$04
STA $01
CLI
```

Oppure, in Basic:

```
Poke 1, Peek(1) or 4
Poke 56334, Peek(56334) or 1
```

I passi da compiere sono quindi i seguenti:

1- Disabilitare l'interrupt
2- Abilitare la ROM del generatore
3- Leggere la ROM secondo la formula vista prima.
4- Disabilitare la ROM
5- Riattivare l'interrupt.

Nel caso in cui queste operazioni vengano fatte da Basic è necessario eseguirle tramite programma, perchè se sono impartite in modo diretto provocano l'inchiodamento della macchina.

Gli indirizzi di partenza della ROM sono i seguenti:

$D000 per il set maiuscolo / grafico
$D800 per il set minuscolo / maiuscolo

```
Bit: 76543210
     ........   1' Byte
     111111..   2' Byte
     11......   3' Byte
     11......   4' Byte
     11111...   5' Byte
     11......   6' Byte
     11......   7' Byte
     111111..   8' Byte
```

Codificazione del carattere "E" maiuscolo. Si noti la corrispondenza tra i punti (pixel spenti) e gli "1" (punti accesi) che verranno visualizzati sullo schermo grafico.

Disassemblato commentato

(Rilocato da $C000)

By Mariani Giancarlo

```
c000 lda #$00      ;Default x sovrapposi
c002 sta $02       ;(=0).
c004 lda #$00      ;Default per pagina
c006 sta $f9       ;grafica ($E000)
c008 lda #$e0      ;
c00a sta $fa       ;
c00c lda #$00      ;Memoria di schermo
c00e sta $fd       ;($0400)
c010 lda #$04      ;
c012 sta $fe       ;
c014 lda #$2c      ;Controlla la 1'
c016 ldy #$00      ;virgola: se non c'e'
c018 cmp ($7a),y   ;
c01a bne $c039     ;salta avanti.
c01c jsr $aefd     ;Salta la virgola.
c01f jsr $b79e     ;Prende parametro.
c022 stx $02       ;Lo mette in $02.
c024 lda #$2c      ;Controlla la seconda
c026 ldy #$00      ;virgola: se non c'e'
c028 cmp ($7a),y   ;
c02a bne $c039     ;salta avanti.
c02c jsr $aefd     ;Salta la II virgola.
c02f jsr $ad8a     ;Prende parametro a
c032 jsr $b7f7     ;16 bit.
c035 sty $f9       ;Lo mette in $F9/$FA
c037 sta $fa       ;(Pagina grafica).
c039 ldx #$d0      ;Controlla se il 64
c03b lda $d018     ;e' in maiuscolo o
c03e and #$02      ;minuscolo e setta
c040 beq $c044     ;l'inizio del generat.
c042 ldx #$d8      ;di caratteri alla
c044 stx $f7       ;giusta locazione.
c046 lda #$00      ;Generatore di carat..
c048 sta $fb       ;($D000-maiuscole,
c04a lda $f7       ;$D800-minuscole).
c04c sta $fc       ;
c04e ldy #$00      ;Carica un byte dalla
c050 lda ($fd),y   ;memoria di schermo.
[illegible]        ;
c053 beq $c064     ;
c055 clc           ;Calcola l'indirizzo
c056 lda $fb       ;della locazione del
c058 adc #$08      ;generatore di carat-
c05a bcc $c05f     ;teri da cui partire
c05c inc $fc       ;a leggere il caratt.
c05e clc           ;preso dallo schermo.
c05f sta $fb       ;
c061 dex           ;
c062 bne $c056     ;
c064 ldy #$00      ;
c066 sei           ;Disabilita interrupt.
c067 lda $01       ;Abilita la lettura
c069 and #$fb      ;del generatore di
c06b sta $01       ;caratteri.
c06d lda ($fb),y   ;Legge un byte dal gen
c06f sta $ff       ;di car. e salva in FF
c071 lda $02       ;Controlla la sovrap-
c073 beq $c07f     ;posizione.
c075 lda #$35      ;Se c'e', legge il
c077 sta $01       ;byte dalla pagina
c079 lda ($f9),y   ;grafica e lo
c07b ora $ff       ;sovrappone a quello
c07d sta $ff       ;del gen. di caratt.
c07f lda #$37      ;Ripristina la
c081 sta $01       ;locazione 1.
c083 cli           ;Abilita interrupt.
c084 lda $ff       ;Carica il byte
c086 sta ($f9),y   ;Lo mette in pag.graf.
c088 iny           ;Incrementa il cont.
c089 cpy #$08      ;e' arrivato a 8?
c08b bne $c066     ;No: ripeti.
c08d lda #$08      ;Si: Incrementa le
c08f clc           ;locazioni per la
c090 adc $f9       ;pagina grafica di
c092 bcc $c096     ;8 volte.
c094 inc $fa       ;
c096 sta $f9       ;
c098 inc $fd       ;Incrementa le locaz.
c09a bne $c09e     ;per la memoria di
c09c inc $fe       ;schermo.
c09e lda $fe       ;Sono arrivate a
c0a0 cmp #$07      ;$F7E8 (2023)?
c0a2 bne $c046     ;
c0a4 lda $fd       ;
c0a6 cmp #$e8      ;
c0a8 bne $c046     ;No: ripeti
c0aa rts           ;Si:torna al basic.
```

```
1000 PRINT CHR$(147);"TRASFERISCE TESTO IN GRAFICA":PRINT
1010 PRINT "USO: SYS XXXX [,OPZ [,PAG]]":PRINT
1020 PRINT"XXXX= INDIRIZZO DI PARTENZA QUALSIASI":PRINT
1030 PRINT"OPZ= OPZIONE: 0=CANCELLA, 1=SOVRAPPONE      (OPZIONALE:DEF=0)":PRINT
1040 PRINT"PAG= INDIRIZZO PAGINA GRAFICA (OPZIONALE      DEFAULT=57344=$E000)"
1050 RETURN
1060 :
1100 DATA 169,000,133,002,169,000,133,249,169,224,133,250,169,000,133,253,169
1110 DATA 004,133,254,169,044,160,000,209,122,208,029,032,253,174,032,158,183
1120 DATA 134,002,169,044,160,000,209,122,208,013,032,253,174,032,138,173,032
1130 DATA 247,183,132,249,133,250,162,208,173,024,208,041,002,240,002,162,216
1140 DATA 134,247,169,000,133,251,165,247,133,252,160,000,177,253,170,240,015
1150 DATA 024,165,251,105,008,144,003,230,252,024,133,251,202,208,242,160,000
1160 DATA 120,165,001,041,251,133,001,177,251,133,255,165,002,240,010,169,053
1170 DATA 133,001,177,249,005,255,133,255,169,055,133,001,088,165,255,145,249
1180 DATA 200,192,008,208,217,169,008,024,101,249,144,002,230,250,133,249,230
1190 DATA 253,208,002,230,254,165,254,201,007,208,162,165,253,201,232,208,156
1200 DATA 096,-1,24698
```

PRINCIPIANTI, I DIECI COMANDAMENTI

1 Leggi attentamente il libretto di istruzioni del tuo computer e delle periferiche che ti sei procurato (registratore, drive, stampante, monitor, eccetera)

2 Non limitarti a leggere i listati di esempio ivi riportati, anche se sembrano banali: digitali sul computer e falli girare (digitando Run e premendo il tasto Return). Se non ti sono chiari, al contrario, la loro digitazione (e successiva esecuzione) ti chiarirà le idee.

3 Nel digitare i listati, ricordati di premere SEMPRE il tasto Return quando giungi alla fine del rigo, anche se tale operazione può sembrarti inutile.

4 Non confondere la vocale alfabetica "O" con il numero zero "0".

5 Digita sempre per esteso il comando PRINT e non abbreviarlo mai con il punto di domanda (?).

6 Inizia a digitare i listati più semplici e brevi: quelli più lunghi potrai digitarli quando avrai acquisito una maggior dimestichezza con il computer.

7 Dopo aver digitato un QUALSIASI programma, registralo subito, seguendo le istruzioni riportate sul manuale, PRIMA di dare Run.

8 Dopo aver fatto partire un programma, in caso, ad esempio, di segnalazione di errore in linea 350, digita soltanto...

List 350

...e accertati che la linea che appare sul video sia RIGOROSAMENTE identica a quella stampata sulla rivista.

9 Se, digitando un listato, ti accorgi che vi sono istruzioni di tipo Poke e Sys, raddoppia la prudenza nella digitazione e nelle procedure di registrazione PRIMA di dare Run.

10 Accertati che il programma che ti accingi a digitare sia REALMENTE valido per il tuo computer. Su ogni articolo della nostra rivista (da leggere SEMPRE con attenzione) è indicato il tipo di computer per il quale il listato stesso è idoneo.

ELEMENTARE, WATSON!

In questo articolo ci proponiamo di aiutare a risolvere alcune fra le migliori "avventure" finora proposte dalla Systems Editoriale

di **Marco Corazza e Michele Maggi**

Come i lettori più affezionati avranno notato, la Systems Editoriale ha pubblicato alcune avventure, in italiano, particolarmente curate, ma non di immediata soluzione (altrimenti, che gusto c'è?).

Ci riferiamo, in particolare, a "Mission Twain", "Zagor", "Terrore a Dunwich" e "Le notti della luna piena".

In questo articolo aiuteremo concretamente i tanti "avventurieri" in difficoltà, fornendo, tra l'altro, alcune mappe.

La soluzione viene data volutamente in modo discorsivo e superficiale, per non rubare del tutto lo spirito di avventura.

Siete pronti? Avete staccato il telefono, il campanello di casa e siete chiusi in camera vostra?

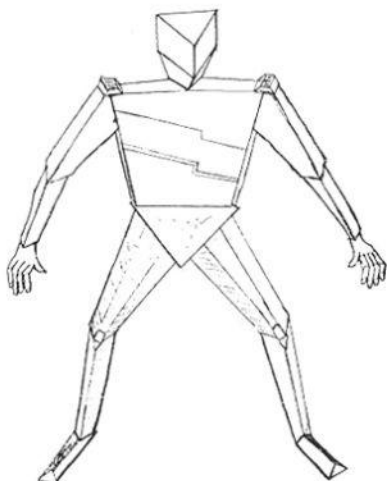
Bene, appendete alla porta il cartello "Non disturbare" e, dopo aver saccheggiato la dispensa, siate pronti a cominciare...

MISSION TWAIN

Anno 2154. Un'astronave sta viaggiando nello spazio profondo alla ricerca di qualcosa o qualcuno. E' una missione scientifica, è Mission Twain.

L'approdo ad un pianeta sconosciuto è solo all'inizio un evento auspicato: da quel pianeta nessuno uscirà vivo.

Ma non precipitiamo troppo le cose; i nostri collaboratori, e perfino il Comandante, sono scomparsi; decidiamo quindi di uscire dall'astronave per cercarli.

All'inizio ci troviamo in una radura con un enorme albero posto proprio dinanzi a noi. Salire sull'albero per vedere cosa ci circonda sarebbe una buona idea, peccato però che non sia possibile... almeno così pare. In realtà occorre procurarsi le scarpe chiodate.

Dalla cima dell'albero si vedono molte cose interessanti e conviene anche dare un'occhiata tra i rami.

Dirigendosi verso sud si arriva alla palude ove molti sprofondano inesorabilmente.

Per sopravvivere sarà meglio alleggerirsi di indumenti ormai inutili.

Intorno alla palude troviamo una radura silenziosa in cui sono celati numerosi oggetti interessanti.

Sempre nei dintorni, troviamo una grotta usata come tana da qualche essere misterioso; per entrare occorre spostare il macigno che nasconde l'ingresso.

Durante il cammino incontriamo una pianta carnivora gigante che nasconde un mantello (il burnus), utile per mimetizzarsi quando incontreremo il Ragnotermite. Giungiamo così alla cupola, proprio quella che si vedeva dalla cima dell'albero. Per entrarvi occorre utilizzare un particolare oggetto, ma per resistere ai raggi colorati (che ci investiranno poco dopo) occorre essere in forze (il comando "score" vi permette di conoscere le vostre condizioni).

Ci troviamo ora teletrasportati all'interno di una base spaziale militare. Scopriamo abbastanza facilmente che esaminando il libro (nello studio) compare una stanza segreta in cui è riposto un telecomando.

Più difficile è, invece, accorgersi dei raggi mortali che impediscono di percorrere il corridoio; se riusciremo ad evitarli arriveremo in una stanza circolare. Da qui prendiamo la scaletta e poi, sempre strisciando, usciamo dalla porta situata a ovest.

Ci troviamo ora in una serra... le piante nascondono un pannello luminoso fornito di tastiera su cui è visualizzato "Enter password:". Ricordate quello che abbiamo letto nella grotta?

La nostra avanzata prosegue e nemmeno il robot-droide di guardia ci potrà fermare; per eliminarlo usiamo la pistola laser. La cella (chiusa) si apre con il telecomando e all'interno troviamo... il Comandante! Dopo tanto tempo possiamo scambiare due parole con un essere umano. Ma non c'è tempo da perdere ed il Comandante freme per fuggir via. Ricordiamoci di abbassare la leva, prima di salire sulla navicella, e di avere a disposizione il carburante necessario perchè vorrà dire che avremo completato la missione.

Cara, dolce terra, stiamo tornando...

Mission Twain verrà ripubblicato, a grande richiesta, sul dischetto "Commodore 64 Club" N. 8.

ZAGOR

Nelle vesti di Zagor, lo spirito con la scure, dovremo recuperare una formula Top Secret caduta nelle mani del temibile Smirnoff.

A chiedere il nostro intervento sono Raffer e Rodolfo, alti emissari dello stato di Badenlandia, cui la formula è stata sottratta. Chieste, ed ottenute, le mappe con la dislocazione del malvagio Smirnoff, presa la scure, iniziamo la nostra avventura al fianco di Cico, insaziabile ed inseparabile amico.

Giunti a Moraca ci riforniamo di provviste nell'emporio e ci procuriamo un carro con i cavalli.

Proprio in questo paese vive il conte Lapalette, vecchio compagno di avventure, le cui doti sono legate alla capacità di sottrarre denaro ai poveri malcapitati. Solo un tipo geniale come lui potrà aiutarci ad eludere i vari sistemi di sicurezza una volta entrati nel castello di Smirnoff.

Gli raccontiamo la storia e gli mostriamo le mappe; naturalmente accetta di partecipare all'avventura e la sua mente ha già elaborato un congegno diabolico per recuperare la formula Top Secret.

Il viaggio è lungo e il crepuscolo ci costringe a fermarci per trascorrere la notte.

Il mattino seguente proseguiamo il cammino finchè non arriviamo nei pressi della fortezza. Nel bosco troviamo un rampone (che ci sarà utile); in seguito incontriamo un villaggio di indiani Sequin.

Rendiamo omaggio allo stregone e gli raccontiamo della missione.

Salutato e ringraziato lo stregone, giungiamo nei pressi di un pianoro, leghiamo la corda al rampone e ci arrampichiamo; siamo davanti al forte. Neutralizziamo la guardia sulla torretta e, servendoci della corda, entriamo nel forte.

Immobilizzata la guardia entriamo nella palazzina dove si dovrebbe trovare la cassaforte. Una volta entrati nello studio di Smirnoff occorre trovare il passaggio per raggiungere la camera blindata. Non è facilissimo: bisogna fare in modo di neutralizzare i pesci pirana per esaminare l'acquario.

Questo nasconde una scalinata (in discesa). Siamo ora davanti alla camera blindata; invitiamo il conte a fare la sua parte e, dopo alcuni interminabili minuti, si riesce a recuperare la formula.

Bisogna ora fuggire senza farsi scoprire; presa la formula percorriamo il cammino a ritroso fino all'esterno della palazzina. Individuato il portone principale, neutralizziamo la guardia.

Una volta fuori dal forte raggiungiamo il carro e torniamo a Moraca.

La prossima tappa è la capanna.

Ad attenderci ci sono Raffer e Rodolfo, cui consegnamo la formula.

Giustizia è fatta.

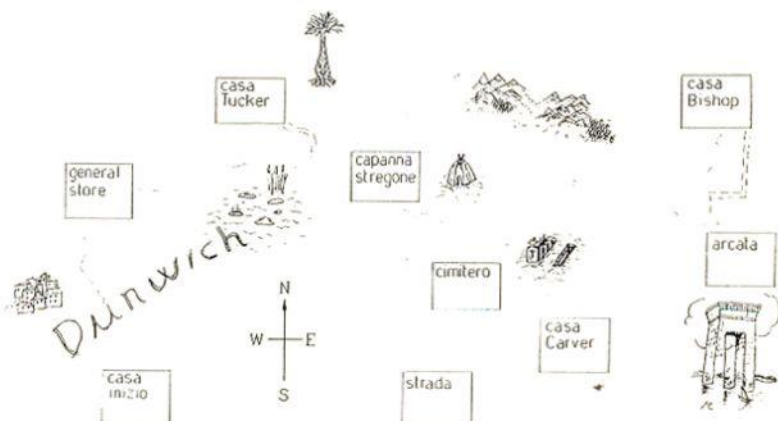
TERRORE A DUNWICH

Questa avventura si svolge in Inghilterra e precisamente nella periferia di Londra, dove abbiamo affittato una casa per trascorrere le vacanze. L'unico impegno, se così vogliamo chiamarlo, è quello di visitare una persona per conto di un nostro vecchio amico, il dottor Abraxas.

Dopo un lungo viaggio arriviamo nella casa al termine del pomeriggio; finchè ci sistemiamo giunge la sera; da quello che abbiamo visto la zona è molto tranquilla, per non dire desolata.

All'inizio del gioco siamo nel salotto, mentre fuori infuria un terribile temporale. Alla televisione trasmettono un film horror e l'atmosfera della casa è stranamente suggestiva; accendiamo la candela, prima che l'elettricità venga interrotta, e andiamo in camera da letto.

Ci addormentiamo presto, ma ci svegliamo con un urlo verso mezzanotte: un incubo! Per evitare di passare la notte in bianco sarebbe meglio trovare un sonnifero o un tranquillante in modo da dormire fino al mattino seguente.

Usciamo di casa e troviamo (nella cassetta delle lettere) un messaggio per noi: è di Abraxas, che ci saluta e ci fornisce le indicazioni per rintracciare Tucker. Saliamo perciò sulla macchina e ci dirigiamo verso Dunwich. Del paese rimangono poche case abbandonate ed un General Store.

Entriamo per chiedere informazioni e, dopo aver insistito a lungo, convinciamo il proprietario, tale signor Jenkins, ad accompagnarci alla casa di Tucker.

Pare, infatti, che il nostro uomo sia un tipo un po' solitario e viva nel bel mezzo di una palude.

Arrivati a destinazione ci aspetta una delusione: la casa di Tucker sembra abbandonata e del padrone non c'è traccia visibile. All'interno della casa troviamo una biblioteca piena di libri di magia e di scienze esoteriche. Sulla scrivania scopriamo anche un diario nel quale molte pagine sono dedicate ad un magico albero della vita capace di garantire l'immortalità.

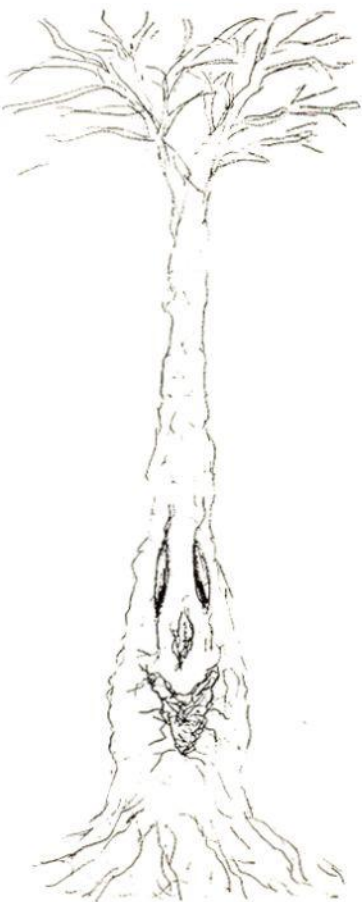
Un po' agitati usciamo dalla casa e cerchiamo Jenkins per tornare indietro, ma dell'uomo non c'è traccia: scomparso anche lui.

Giriamo intorno alla casa e qui abbiamo una visione agghiacciante: Jenkins è imprigionato da un albero gigantesco che lo sta "assorbendo".

In preda al panico ci gettiamo in una pazza corsa, senza sapere bene dove andare; dopo un po' ci mancano le forze e cadiamo a terra svenuti.

Ci raccoglie e ci cura uno stregone che vive nella palude; da lui apprendiamo una incredibile realtà: la morte di Jenkins, risucchiato dall'albero, riporterà alla vita uno zombi che per "sopravvivere" ucciderà chiunque e distruggerà qualunque cosa.

Per far ritornare nel regno delle tenebre questo essere dobbiamo recuperare un libro di magia bianca che un certo Bishop, tempo addietro, ha sottratto allo stregone. Prima di accomiatarci, lo stregone ci indica la strada da seguire e fissa un appuntamento ai piedi dell'arcata impossibile.

Sconcertati, ma decisi, usciamo dalla capanna finchè non ci troviamo sulla strada principale. Senza sapere bene cosa fare, entriamo nella prima casa che troviamo

per chiedere aiuto a qualcuno. In casa Carver troviamo però i cadaveri dei due coniugi: lo zombi è già entrato in azione!

Recuperato un accendino, usciamo di casa e, guidati dal nostro istinto, percorriamo un tratto di strada finchè non incontriamo un'altra casa. Risulta disabitata e in cattive condizioni ma è proprio quella che cerchiamo: la casa di Bishop. Dalle lettere e dai libri abbiamo conferma che Bishop fa parte di una setta mistica dedita alla magia nera. Trovati alcuni oggetti, desideriamo esplorare la casa; ma queste vecchie stamberghe sono sempre piene di botole e di passaggi segreti...

Con un po' di fortuna arriviamo in una cripta in cui troviamo il libro che cerchiamo. Il tunnel sbuca all'aperto nelle vicinanze di un fiume. E' qui che vediamo l'arcata centrale di un ponte sostenuta da tre piloni invece che due: è l'arcata impossibile.

Ad attenderci c'è, infatti, lo stregone. Siamo vicini alla fine; preso il libro, e tracciato un cerchio per terra, lo stregone ci invita ad entrare nel cerchio mentre inizia il rito magico. Poco dopo arriva lo zombi che, varcato il cerchio nel tentativo di aggredirci, si sfalda riducendosi in pochi frammenti di polvere e ossa.

L'incubo è finito.

Anche Terrore a Dunwich, come Zagor e Mission Twain, verrà pubblicato nuovamente in futuro.

SEGUI L'AVVENTURA

Il genere Adventure, ultimamente, risulta particolarmente apprezzato dalle nuove generazioni sia perchè, ormai, i giochi di azione in cui è necessario uccidere marzianetti sono quasi tutti eguali tra loro, sia perchè è stimolante la ricerca (intelligente) di soluzioni cervellotiche.

Non tutti, però, sono in grado di risolvere le varie avventure, e lo testimoniano le numerose lettere e telefonate che giungono in Redazione a proposito dei nostri prodotti.

Di tanto in tanto, quindi, verranno pubblicati alcuni "aiuti" che non devono essere considerati vere e proprie soluzioni dei giochi: sarebbe come comunicare il nome dell'assassino a chi si accinge a leggere un libro giallo. Le note che leggerete, pertanto, si limiteranno a fornire qualche suggerimento (a volte banale, a volte no) che vi metterà in grado di superare gli scogli più difficili.

Al resto, ovviamente, dovrete pensare voi stessi!

LE NOTTI DELLA LUNA PIENA

Dylan Dog, l'indagatore dell'incubo, è appena arrivato in Germania dove, aiutato da Groucho, dovrà risolvere un caso misterioso molto, molto più pericoloso del previsto...

Una decisione fondamentale è quella che si deve prendere all'inizio del gioco: se svoltiamo a sinistra saremo molto avvantaggiati nella risoluzione.

Una volta giunti nella foresta, sarà bene prestare attenzione a tutto ciò che si muove intorno a noi, eventualmente scendendo dalla macchina e curiosando per il bosco.

Capiterà (capita tutti i giorni) di subire attacchi da lupi o altri animali, ma non sempre la via della violenza è quella giusta...

Proseguendo per il bosco è molto probabile trovare un capanno nel quale si cela un'arma determinante per la soluzione dell'avventura.

In un modo o nell'altro si giunge comunque al college dove si fa la conoscenza di Frau Blucher, la direttrice, personaggio quantomai ambiguo e pericoloso.

Evitando (se possibile) di irritarla, le parleremo, ma finiremo inevitabilmente fuori dal college dove, sotto la pioggia, dovremo raggiungere la città più vicina, purtroppo a piedi.

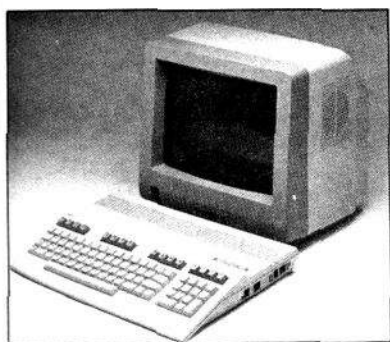
Una volta a Wolfburg sarà determinante la scelta della strada da seguire e la successiva scelta della locanda... attenzione quindi!

Se sopravviverete alla notte verrete svegliati da Alex, una dolce fanciulla con qualche serio problema.

Da lei apprendete molte importanti notizie prima che qualcuno (o qualcosa) ponga prematuramente termine alla sua giovane vita.

A questo punto non resta molto tempo, bisogna tornare alla taverna e fare il punto della situazione assieme all'oste e a Groucho.

Vi attende ora la parte più difficile dell'avventura; dovete riuscire a sconfiggere la strega e ci riuscirete solo se siete in possesso di una certa arma (ricordate il capanno?).

Forza, non perdetevi d'animo, la strada è ancora lunga...

QUANTO COSTA IL TUO COMMODORE

LISTINO PREZZI HOME COMPUTER

Codice	Prodotto	Prezzo (IVA esclusa)
	Personal Computer	
A 500	Personal Computer 16/32 BIT CPU 512K RAM incorpora un floppy disk drive da 3" 1/2 da 880Kb e un mouse	L. 950.000
A 501	Espansione di memoria per Amiga 500 512Kbytes e orologio	L. 212.500
A 520/1	Modulatore per Amiga 500 permette di collegare Amiga 500 ad un qualsiasi televisore b/n e colori	L. 40.000
A SCART	Cavo collegamento Amiga 500 con TV prescart	L. 27.000
Nuovo C64	Nuovo personal computer CPU 64K RAM computer ad alta risoluzione grafica, 256 combinazioni di colori, sintetizzatore di suono. Collegabile ad un qualsiasi televisore	L. 375.000
1764	Espansione di memoria per C64 256Kbytes	L. 195.000
C 128D	Personal computer CPU 128Kb RAM CPU 128Kbytes espandibile a 512Kbytes, 48Kbytes ROM, basic 7.0. Tastiera separata. Alta risoluzione grafica, 16 colori + 8 sprites. 40 80 (RGB) colonne. Programmabile in CP/M 3.0. Funzionante in modo C64. Floppy disk da 340Kb incorporato.	L. 895.000
1700	Espansione di memoria per C128. 128Kb	L. 165.000
1750	Espansione di memoria per C128. 512Kb	L. 235.000
	Registratori	
1530	Registratore compatibile C64 - C128 - C128D	L. 55.000
	Unità a dischi	
1541 II	*Floppy disk drive 5" 1/4. Unità di memoria di massa, drive singolo, capacità 170Kbytes in linea.* Compatibile con C64	*L. 395.000*
1581	Floppy disk drive 3" 1/2. Unità di memoria di massa, drive singolo da 3" 1/2, capacità 800Kbytes. Compatibile C64 - C128 - C128D	L. 420.000
A 1010	Floppy disk drive esterno 3" 1/2	L. 395.000
1802	Monitor a colori, alta risoluzione, 14" con audio. Collegabile al C64, C128, C128D (40 colonne)	
2080	Monitor a colori alta persistenza alta risoluzione, 14", con audio antiriflesso. Collegabile ad Amiga PC. C64, C128. C128D. 640x400 pixel, 4096 colori	L. 630.000
1084	Monitor a colori. Alta risoluzione, 14", con audio, antiriflesso. Collegabile al C64 - C128 - C128D - Amiga - PC. 640x400 pixel, 4096 colori.	L. 545.000
	Stampanti	
MPS 1200 S	Stampante. 80 colonne, 120 cps, bidirezionale, carta in modulo singolo e trascinamento modulo continuo per C64 - C128 - C128D	L. 480.000

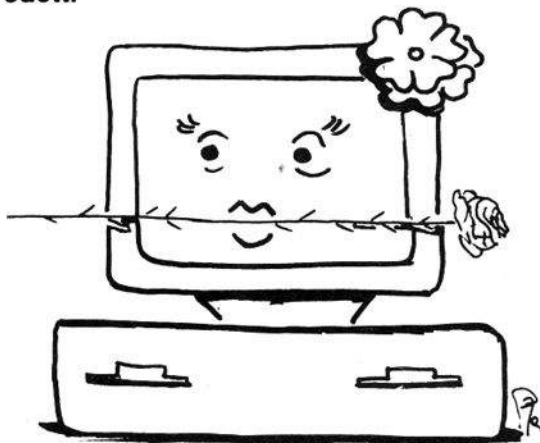
MPS 1200 P	Stampante 80 colonne 120 cps, bidirezionale, carta in modulo singolo e trascinamento modulo continuo per A500	**L. 480.000**
MPS 1250	Stampante con stesse caratteristiche del modello MPS 1200 P, interfacce seriale RS232C e parallela Centronics di serie	**L. 495.000**
MPS 1500	Stampante a colori. 80 colonne, 130 cps, 4 colori, bidirezionale, carta in modulo singolo e trascinamento modulo continuo.	**L. 550.000**
	Accessori	
6499	Adattatore telematico omologato. Collegabile al C64, permette il collegamento a Videotel PGE-Banche Dati.	**L. 149.000**
1351	Mouse. Per C64, C128, C128D	**L. 99.000**
1311	Joystick. Comando per giochi	**L. 13.500**
1399	Joystick. 8 microswitch-autofire	**L. 29.000**

LISTINO PREZZI SISTEMI

Codice		**Prezzo (IVA esclusa)**
PCI	Microprocessore 8088 (coprocessore 8087 opzionale) frequenza clock 4,77MHz. RAM 512Kb. Espandibile a 640Kb; 1 floppy da 360Kb. Monitor a fosfori verdi 12".	**L. 945.000**
PC 10-III	Microprocessore Intel 8088 a 16 bit; clock 10 MHz; memoria RAM: 640 Kb; memoria ROM (BIOS): 8 Kb autoconfigurabile; sistema operativo MS-DOS 3.21; GW-Basic; scheda video integrata monocromatico/colore di serie con emulazioni CGA/MDA/Hercules via software; 2 floppy disk drive da 360 Kb; monitor monocromatico a fosfori verdi da 12"; porte seriale RS232C e paralella Centronics; tastiera avanzata 102 tasti con 12 tasti funzione; porta mouse, incluso controller per H.D.	**L. 1.490.000**
PC 10-IIIC	Stessa configurazione ma con monitor 14" a colori mod. 1084	**L. 1.890.000**
PC 20-III	Microprocessore Intel 8088 a 16 bit; memoria RAM da 640Kb; memoria ROM (BIOS) da 8Kb autoconfigurabile. Sistema operativo MS-DOS 3.21. Scheda video monocromatico/colore di serie con emulazioni CGA/MDA/Hercules via software. Un hard disk da 20Mb e un floppy disk drive da 360Kb. Monitor monocromatico a fosfori verdi da 12". Porta seriale RS232C parallela Centronics. Porta mouse incluso controller per H.D.; tastiera avanzata 102 tasti con 12 tasti funzione	**L. 2.240.000**
PC 20-IIIC	Stessa configurazione ma con monitor 14" a colori mod. 1084	**L. 2.640.000**
PC 40/20 AT	Microprocessore 80286 a 16 bit; clock 10/6 MHz; memoria RAM dal 1 Mb; hard disk da 20 Mb; un floppy disk drive da 1,2 Mb; Sistema operativo MS-DOS 3.2, GW-Basic; scheda video monocromatico/colore 132 colonne AGA di serie; monitor monocromatico a fosfori verdi da 14"; porta seriale RS232C e parallela Centronics; tastiera avanzata 102 tasti con 12 tasti funzione	**L. 3.490.000**
PC 40/40 AT	Microprocessore 80286 a 16 bit; clock 10/6 MHz; memoria RAM da 1Mb. Un hard disk da 40Mb; un floppy disk drive da 1,2Mb. Sistema operativo MS-DOS3.2, GW-Basic. Scheda video monocromatico/colore EGA WONDER di serie. Monitor monocromatico a fosfori verdi da 14". Porta seriale RS232 e parallela Centronics; tastiera avanzata 102 tasti con 12 tasti funzione	**L. 4.490.000**
PC 60/40	Microprocessore 80386 a 32 bit; clock 8/16 MHz; memoria RAM da 2,5 MB; hard disk da 40 Mb; 1 floppy disk drive da 1,2 Mb; sistema operativo MS-DOS 3.20; scheda EGA WONDER di serie; monitor ADI monocromatico a fosfori verdi da 14"; 2 porte seriali RS232C e 2 parallele Centronics. In opzione: floppy disk drive da 3" 1/2 e 1.44 Mb. Tastiera avanzata 102 tasti con 12 tasti funzione	**L. 8.490.000**
PC 60/80	Stessa configurazione del PC 60/40 ma con hard disk da 80 Mb e, in più, MS-DOS WINDOWS 386 e mouse	**L. 9.990.000**

Amiga 2000	*Microprocessore Motorola 68000; clock 7,16 MHz; 1MB RAM; 256KB ROM; kickstart in* ROM. Uscita PAL; 4 uscite a 4 voci su 2 canali (stereofonia). Amiga DOS; Amiga multi-tasking. 7 slots di sistema (2 combinati Amiga - Ibm PC AT); 1 slot video; 1 accesso diretto CPU 86 pin. Compatibilità MS-DOS interna con schede Janus (XT/AT CARD). Possibilità di un secondo drive interno da 3" 1/2 e di un drive interno da 5" 1/4. Monitor a colori 1084 tastiera; mouse. Workbench; Extras. MONITOR ESCLUSO	**L. 2.005.000**
	Monitor	
2080	*Monitor a colori ad alta risoluzione e lunga persistenza 14" con audio; 640x400 pixel; 4096* colori per A500/A1000/A2000/PC AT	**L. 630.000**
1084	Monitor a colori ad alta risoluzione 14" con audio; antiriflesso; 640x400 pixel; 4096 colori per A500/A1000/A2000/PC AT	**L. 545.000**
1402	Monitor monocromatico da 12"	**L. 255.000**
1450	Monitor monocromatico BISYNC	**L. • □**
1950	Monitor colore BISYNC	**L. • □**
A 2024	Monitor monocromatico alta risoluzione a pagina bianca	**L. • □**
	Stampanti	
MPS 1200P	Stampante a 9 aghi 120 cps bidirezionale 80 colonne. Interfaccia Commodore per PC 10/III PC 40 PC/60/80	**L. 480.000**
MPS 1250	Stesse caratteristiche del modello MPS 1200P, interfacce seriale RS232C e parallela Centronics di serie	**L. 495.000**
MPS 1500	Stampante a colori a 9 aghi 130 cps; bidirezionale; 80 colonne; carta in modulo singolo e trascinamento modulo continuo; 4 colori; per Amiga e PC	
	Unità dischi	
K 910	Floppy disk drive aggiuntivo da 3" 1/2 e 1.44 Mb per PC 10-III e PC 20-III	**L. 295.000**
A 1010	Floppy disk drive esterno da 3" 1/2	**L. 395.000**
A 2010	Floppy disk drive interno da 3" 1/2	**L. 290.000**
A 2092 + PC 5060	Hard disk da 20 Mb settorizzabile PC/Amiga + scheda controller per hard disk per A 2000	**L. 845.000**
A 2080 + A 2020	Scheda Janus XT compatibile + drive	**L. 895.000**
A 2286 + A 2020	Scheda Janus II AT compatibile + drive	**L. 1.285.000**
A 2620	*Scheda 68020 con processore a 32 bit; 14,28 MHZ*	**L. •**
A 2090 + A 2092	Hard disk 20 Mb; interfaccia SCSI per A 2000	**L. 1.025.000**
A 2090 + A 2094	Hard disk 40 Mb; interfaccia SCSI per A 2000	**L. 1.575.000**
A 590	Hard disk da 20 Mb + fast RAM per A 500	**L. • □**
	Accessori	
PC EXP1	Box esterno per espansione PCI; può contenere 3 schede aggiuntive e hard disk alimentazione e cavi compresi	**L. 590.000**
A 2052	Scheda di espansione RAM da 2 Mb per A 2000 (max 4)	**L. 590.000**
A 2060	Scheda video/RF molulatore per TV e VCR per A2000	**L. 138.000**
A 2058	Espansione di memoria da 8 Mb RAM per A2000	**L. 3.760.000**
A UNIX	Scheda UNIX utilizzabile con A2620 (68020 board)	**L. • □**
1352	*Mouse due tasti collegabile alla serie dei PC include Microsoft Mouse Driver*	**L. 72.000**
A 2300	Genlock card semiprofessionale per A2000; permette di miscelare grafici con videocamere VCR; selettore per video Amiga, monitors esterni, proiettori; include software per titolazione	**L. 345.000 Δ**
A 2350	Professional Video Adapter Card per A 2000. Genlock professionale con freeze frame e digitalizzatore video incluse software di controllo per la gestione interattiva	**L. 1.645.000 Δ**

• prezzo da definire
Δ disponibile da giugno '88
□ disponibile da settembre '88

I primi 100 negozi selezionati dalla "Commodore" per la vendita dei suoi prodotti

Al Risparmio	*V.le Monza 204*	20128 Milano	Tel. 02-2573440
Braha Alberto S.D.F.	*Via Pier Capponi 5*	20145 Milano	Tel. 02-437468
E.D.S.	*C.so Porta Ticinese 4*	20123 Milano	Tel. 02-8322045
ESC	*Via Roggia Scagna 7*	20127 Milano	Tel. 02-2871300
Faref	*Via A. Volta 21*	20121 Milano	Tel. 02-650042
Giglioni	*Via D'Ovidio 8*	20131 Milano	Tel. 02-2360397
Giglioni	*V.le Luigi Sturzo 45*	20154 Milano	Tel. 02-654906
Logitek	*Via Golgi 60*	20133 Milano	Tel. 02-538931
Marcucci	*Via F.lli Bronzetti 37*	20129 Milano	Tel. 02-7386051
Melchioni	*Via P. Colletta 37*	20135 Milano	Tel. 02-57941
Messaggerie Musicali	*Galleria Del Corso*	20100 Milano	Tel. 02-50841
Newel	*Via Mac Mahon 75*	20155 Milano	Tel. 02-323492
Rivola	*Via Vitruvio 43*	20124 Milano	Tel. 02-6694160
F.lli Galimberti	*Via Naz. dei Giovi 28/36*	20030 Barlassina (Mi)	Tel. 0362-560625
P. Giorgio Ostellari	*Via Molino Arese 65*	20031 Cesano Mad.(Mi)	Tel. 0362-504392
P. Giorgio Ostellari	*Via Milano 300*	20233 Desio (Mi)	Tel. 0362-621042
GBC Italiana	*V.le Matteotti 66*	20092 Cinisello B. (Mi)	Tel. 02-6181801/6189391
	Via Petrella 6	20100 Milano	Tel. 02-203607/8
	Via Cantoni 7	20100 Milano	Tel. 02-437478
Casa Della Musica	*Via Indipendenza 21*	20093 Cologno Mon.(Mi)	Tel. 02-2542117
Penati	*Via Verdi 28/30*	20011 Corbetta (Mi)	Tel. 02-9779401
EPM System	*V.le Italia 12*	20094 Corsico (Mi)	Tel. 02-4407979
Centro Comp. Pandolfi	*Via Corridoni 18*	20025 Legnano (Mi)	Tel. 0331-546426
Computeam	*Via Vecellio 41*	20035 Lissone (Mi)	Tel. 039-481010
Futura	*Via Solferino 31*	20075 Lodi (Mi)	Tel. 0371-54457
MBM Inf. Sys.	*Corso Roma 112*	20075 Lodi (Mi)	Tel. 0371-53610
L'Amico del computer	*V.le Lombardia 17*	20077 Melegnano (Mi)	Tel. 02-9838341
Bit 84	*Via Italia 4*	20052 Monza (Mi)	Tel. 039-320813
I.C.O.	*Via dei Tigli 14*	20090 Opera (Mi)	Tel. 02-5242146
Comif	*Via Autolinee 10*	24100 Bergamo	Tel. 035-218553
Cordani	*Via dei Caniana 8*	24100 Bergamo	Tel. 035-258184
D.R.B.	*Via Borgo Palazzo 65*	24100 Bergamo	Tel. 035-237292
New Systems	*Via Paglia 36*	24100 Bergamo	Tel. 035-248109
Computer Team hi-tec	*Via Verdi 1/B*	24030 Carvico (Bg)	Tel. 035-790244
Ott. Optometr. Rovetta	*P.zza Garibaldi 6*	24065 Lovere (Bg)	Tel. 035-960705
A.I.S. International	*Via San Carlo 25*	24016 S.Pellegrino T.(Bg)	Tel. 0345-22662
Sisthema	*Via Roma 45*	24067 Sarnico (Bg)	Tel. 035-910750
Computer Center	*Via Cipro 62*	25125 Brescia	Tel. 030-223230
Informatica 2000	*Via Stazione 16/B*	25100 Brescia	Tel. 030-54015
Vigasio Mario	*Portici Zanardelli 3*	25100 Brescia	Tel. 030-59330/295858
Mister Bit	*Via Mazzini 70*	25043 Breno (Bs)	Tel. 0364-22835
Cavalli Pietro	*Via 10 Giornate 14B*	25030 Castrezzato (Bs)	Tel. 030-714013
Vietti Giuseppe	*Via Milano 1/B*	25032 Chiari (BS)	Tel. 030-7100206
Megabyte	*P.zza Duomo 17*	25015 Desenzano G.(Bs)	Tel. 030-9144880
Ditta Resi Rino	*Via XX Settembre 7*	25016 Ghedi (BS)	Tel. 030-901224
Info Cam	*Provinciale 38*	25050 Gratacasolo (Bs)	Tel. 0364-89379
Il Computer	*Via Indipendenza 90*	22100 Como	Tel. 031-240959
2M Elettronica	*Via Sacco 3*	22100 Como	Tel. 031-278227
Eletrongros	*Via L. Da Vinci 54*	22062 Barzanò (Co)	Tel. 039-957318
Ega	*Via Mazzini 42*	22065 Cassago Br.(Co)	Tel. 039-956307
Ega	*Via Aldo Moro 17*	22043 Galbiate (Co)	Tel. 0341-522028
Data Found Comp. Sh.	*Via A. Volta 4*	22036 Erba (Co)	Tel. 031-645761
Righi Elettronica	*Via G. Leopardi 26*	22077 Olgiate Com. (Co)	Tel. 031-946766
Fumagalli	*Via Cairoli 48*	22053 Lecco (Co)	Tel. 0341-863341
Cima Eelettronica	*Via Leonardo Da Vinci 7*	22053 Lecco (Co)	Tel. 0341-371106
Mondo Computer	*Via Giuseppina 11/B*	26100 Cremona	Tel. 0372-882079
Prisma	*Via Buoso Da Dovara 8*	26100 Cremona	Tel. 0372-437900
Telco	*P.zza Marconi 2/A*	26100 Cremona	Tel. 0372-31544
Elcom/GBC	*Via IV Novembre 56/58*	26013 Crema (Cr)	Tel. 0373-83393
Euroelettronica	*Via XX Settembre*	26013 Crema (Cr)	Tel. 0373-86966
Kofbaker	*Via Marchi 65/B*	26039 Vescovato (CR)	Tel. 0372-818601
Computer	*Galleria Fermi 7*	46100 Mantova	Tel. 0376-325616
32 Bit (Comp. St.)	*Via Cesare Battisti 14*	46100 Mantova	Tel. 0376-326770
Elettronica di Basso	*V.le Risorgimento 69*	46100 Mantova	Tel. 0376-329311
Poliware	*C.so C. Alberto 76*	27100 Pavia	Tel. 0382-22636
Log. Infor. Com. Shop	*V.le Monte Grappa 32*	27029 Vigevano (Pv)	Tel. 0381-81883
M. Visentin	*C.so Vitt. Emanuele 76*	27029 Vigevano (Pv)	Tel. 0381-83833
Computer Line	*Via G. Carducci 4*	29100 Piacenza	Tel. 0523-30691
Delta Computer	*Via Mart. della Resistenza 15/4*	29100 Piacenza	Tel. 0523-753318
Sover	*Via IV Novembre 60*	29100 Piacenza	Tel. 0523-34388
Cipolla Mauro	*Via Tremogge 25*	23100 Sondrio	Tel. 0342-213569
Fotonova		23010 S. Pietro Ber. (So)	Tel. 0342-492319
Dimeco Sistemi	*Via Garibaldi*	21100 Varese	Tel. 0332-237201
Il Centro Elettronico	*Via Morazzone 2*	21100 Varese	Tel. 0332-231006

Supergames	*Via Carrobbio 13*	21100 Varese	Tel. 0332-241092
Busto Bit	*Via Gavinana 17*	21052 Busto Arsizio (Va)	Tel. 0331-625034
Crespi Giuseppe & C.	*V.le Lombardia 59*	21053 Castellanza (Va)	Tel. 0331-503023
Computer Shop	*Via A. Da Brescia 2*	21013 Gallarate (Va)	Tel. 0331-798612
Lima Import-Export	*Via Clerici 196*	21040 Gerenzano (Va)	Tel.
J.A.C. nuove tecn.	*Via Matteotti 38*	21018 Sesto Cal. (Va)	Tel. 0331-923134
Bit Micro	*Via Mazzini 102*	15100 Alessandria	Tel. 0131-443252
West Records	*C.so Roma 85*	15100 Alessandria	Tel. 0131-441090
S.G.E. Elettronica	*Via Bandello 19*	15057 Tortona (Al)	Tel. 0131-867709
Record di Forina G.	*C.so Alfieri 166/3*	14100 Asti	Tel. 0141-34240
Il Diagramma Italia	*P.le Libertà 4*	12100 Cuneo	Tel. 0171-57067
Rossi Computers	*C.so Nizza 42*	12100 Cuneo	Tel. 0171-2339
Punto Bit	*C.so Langhe 26/C*	12051 Alba (Cn)	Tel. 0173-49833
SDI	*Via Vitt. Emanuele 250*	12042 BRA (CN)	Tel. 0172-421392
Aschieri Gianfranco	*C.so Eman. Filiberto 6*	12045 Fossano (CN)	Tel. 0172-62995
Curetti Augusto	*C.so Italia 3*	12054 Mondovì (Cn)	Tel. 0174-42014
Ditta Elettrogamma	*C.so Risorgimento 20*	28100 Novara	Tel. 0321-176358
Elcom	*C.so Mazzini 11*	28100 Novara	Tel. 0321-391293
Programma 3	*V.le Buonarroti 8*	28100 Novara	Tel. 0321-36367/399903
Punto Video	*C.so Risorgimento 391*	28100 Novara	Tel. 0321-477367
Computer	*Via Monte Zeda 4*	28041 Arona (No)	Tel.
All Computer	*C.so Garibaldi 106*	28021 Borgomanero (No)	Tel. 0322-844142
Micrologic	*Via Giovanni XIII 2*	28037 Domodossola (No)	Tel.
Elliott Computer Shop	*Via Don Minzoni 32*	28044 Intra (No)	Tel. 0323-43517
ABA Elettronica	*Via C. Fossati 5/P*	10100 Torino	Tel. 011-302065
Alex Computer e giochi	*C.so Francia 333/4*	10142 Torino	Tel. 011-7730184
C.D.M. Elettronica	*Via Marochetti 17*	10126 Torino	Tel. 011-636345/634900
Computing New	*Via Marco Polo 40/E*	10129 Torino	Tel. 011-501512
De Bug	*C.so Vitt. Emanuele II 22*	10100 Torino	Tel. 011-832986
Desme Universal	*Via San Secondo 95*	10100 Torino	Tel. 011-592551
F.D.S.	*Via Borgaro 86/D*	10100 Torino	Tel. 011-2168900
Computer Home	*Via San Donato 46/D*	10100 Torino	Tel. 011-4731196
Informatica Italia	*C.so Re Umberto 129*	10128 Torino	Tel. 011-501647
MT Informatica	*C.so Giulio Cesare 58*	10100 Torino	Tel. 011-850955/238803
New Bus. Computer	*Via Nizza 45*	10100 Torino	Tel. 011-214235
Radio Tv Mirafiori	*C.so Un. Sovietica 381*	10135 Torino	Tel. 011-616190/6197189
SMT Elettronica	*Via Bibiana 83/B*	10147 Torino	Tel. 011-218243
Paul e Chico Videos.	*Via Vitt. Emanuele 52*	10023 Chieri (To)	Tel. 011-9470295
Bit Informatica	*Via V. Emanuele 154*	10073 Ciriè (To)	Tel. 011-9205455
HI-FI Club	*C.so Francia 92C*	10093 Collegno (To)	Tel. 011-4110256
ICS	*Stradale Torino 73*	10015 Ivrea (To)	Tel. 0125-251797
Cerrutti Mauro	*C.so Torino 234*	10064 Pinerolo (To)	Tel. 0121-70316
Eurex	*C.so Indipendenza 5*	10086 Rivarolo C.se (To)	Tel. 0124-27984
Ditta Elettrogamma	*C.so Bormida*	13100 Vercelli	Tel. 0161-53689
Elettronica	*Strada Torino 15*	13100 Vercelli	Tel. 0161-393163
C.S.I. Teorema	*Via Losana 9*	13051 Biella (Vc)	Tel. 015-28622
Fotostudio Trevisan	*Via XXV Aprile 24/B*	13014 Cossato (VC)	Tel. 015-921431
Studio Fotogr. Imarisio	*P.zza Martiri Libertà 7*	13039 Trino (Vc)	Tel. 0161-82081
F.lli Gatti	*Via Festaz 75*	11100 Aosta	Tel. 0165-35659

PRODOTTI SYSTEMS EDITORIALE

Software su cassetta

La voce III	L.12000
Raffaello	L.10000
Oroscopo	L.12000
Computer-Music	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.16000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.10000

Software su disco

Ms-Dos & Gw-Basic	L.25000
Ms-Dos & Gw-Basic (con prova di acquisto cassetta)	L.15000
La voce III	L.12000
Raffaello	L.10000
Oroscopo	L.12000
Computer-Music	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.24000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.20000
Graphic Expander C/128 in modo 80 colonne	L.27000
Linguaggio macchina + Routine grafiche	L.12000

Offerta "Commodore speciale L.M." + dischetto — L.16000

Libri

64 programmi per il Commodore 64	L. 4800
I miei amici C/16 e Plus/4	L. 7000
Strategie vincenti per Commodore 64	L. 5800
62 programmi per Vic 20, C/16 e Plus/4	L. 6500
Utilities e giochi didattici	L. 6500
Tutti i segreti dello Spectrum	L. 7000
Simulazioni e test per la didattica	L. 7000
Impara giocando il Basic dello Spectrum	L. 7000
Micro Pascal per Commodore 64/128	L. 7000
Dal registratore al drive del C/64	L. 7000
Ada	L. 5000
Il linguaggio Pascal	L. 5000

Directory

Ciascun dischetto	L.12000

Arretrati

Ciascun numero arretrato di Commodore Computer Club	L. 5000
Ciascun numero arretrato di Personal Computer	L. 5000
Ciascun numero arretrato di VR Videoregistrare	L. 5000

Per un ottimale utilizzo del software "Matematica finanziaria" è opportuna la lettura degli articoli relativi pubblicati sui N. 13, 14, 15 della rivista "Commodore" e sui N. 1, 2 e 3 della Rivista Personal Computer.

Per un ottimale utilizzo del software "Analisi di Bilancio" è opportuna la lettura degli articoli relativi pubblicati sui N.2, 3, 5 della Rivista Personal Computer.

Per un ottimale utilizzo del software "Linguaggio Macchina e Routine grafiche per C/64" è opportuna la lettura del fascicolo "Commodore Speciale" appositamente dedicato.

Coloro che desiderano procurarsi i prodotti della Systems Editoriale devono inviare, oltre alla cifra risultante dalla somma dei singoli prodotti, la cifra di L.3000 per spese di imballo e spedizione, oppure L.6000 se si preferisce la spedizione per mezzo raccomandata.

Sconti e agevolazioni

Le spese di imballo e spedizione sono a carico della Systems se ciascun ordine è pari ad almeno L.50000 (di listino).

Gli abbonati hanno diritto allo sconto del 10% e alla spedizione gratuita se la somma totale raggiunge la cifra di L.50000 (di listino).

Oltre alla spedizione gratuita, viene praticato uno sconto del 10% (per gli abbonati è del 20%) se la cifra raggiunta per ciascun ordine raggiunge le L. 100000 (di listino).

Abbonamenti

Commodore Computer Club (11 fascicoli)	L.45000
Personal Computer (11 fascicoli)	L.40000
Commodore Computer Club + Personal Computer (11 + 11 fascicoli)	L.70000
VR Videoregistrare (12 numeri)	L.45000

N.B.: la cifra per gli abbonamenti non può essere conteggiata per ottenere gli sconti e le agevolazioni di cui sopra.

Non è assolutamente possibile inviare materiale contrassegno.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo di recapito telefonico, ma anche il materiale desiderato.

In ogni caso sarebbe opportuno inviare la presente scheda, debitamente compilata, allegando la fotocopia della ricevuta del versamento effettuato.

Chi volesse ricevere più celermente la confezione deve inviare la somma richiesta mediante assegno circolare oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a: Systems Editoriale - Milano.